MetaKP: On-Demand Keyphrase Generation

Anonymous ACL submission

Abstract

Traditional keyphrase prediction methods predict a single set of keyphrases per document, failing to cater to the diverse needs of users and downstream applications. To bridge the gap, we introduce on-demand keyphrase generation, a novel paradigm that requires keyphrases that conform to specific high-level goals or intents. For this task, we present MetaKP, a large-scale benchmark comprising four datasets, 7500 documents, and 3760 goals across news and biomedical domains with human-annotated keyphrases. Leveraging MetaKP, we design both supervised and unsupervised methods, including a multi-task fine-tuning approach and a self-consistency prompting method with large language models. The results highlight the challenges of supervised fine-tuning, whose performance is not robust to distribution shifts. By contrast, the proposed self-consistency prompting approach greatly improves the performance of large language models, enabling GPT-4o to achieve 0.548 SemF1, surpassing the performance of a fully fine-tuned BART-base model. Finally, we demonstrate the potential of our method to serve as a general NLP infrastructure, exemplified by its application in epidemic event detection from social media.

1 Introduction

Keyphrase prediction is an NLP task that has attracted long-lasting research interest (Witten et al., 1999; Hulth, 2003; Meng et al., 2017). Given documents from various domains such as academic writing, news, social media, or meetings, keyphrase extraction and keyphrase generation models output short phrases aiming at encapsulating the key entities and concepts mentioned by the document. Beyond a number of information retrieval applications (Kim et al., 2013; Tang et al., 2017; Boudin et al., 2020), keyphrase prediction methods are widely incorporated into the pipelines of other NLP tasks such as natural language generation (Yao et al., 2019; Li et al., 2020), text summarization (Dou et al., 2021), and text classification (Berend, 2011).

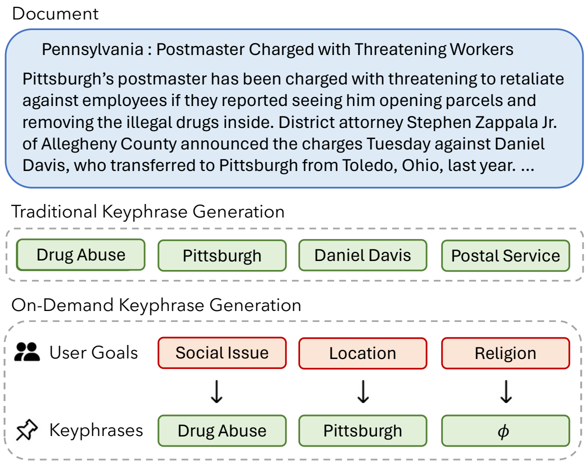

Figure 1: An illustration of on-demand keyphrase generation. Given diverse user goals, models are required to generate goal-conforming keyphrases or abstain.

Despite their wide application in diverse scenarios, which may have diverse requirements on the types of keyphrases, existing keyphrase prediction methods generally follow a suboptimal assumption: for every document, the model shall predict a *single* application-agnostic set of keyphrases, which is then evaluated against a *monolithic* set of references (Wu et al., 2023b). This one-size-fits-all approach fails to cater to both downstream applications' varied requirements of the keyphrase predictions' topic and level of specificity and different expectations from human users with diverse backgrounds. To properly handle such diverse feedback, current approach could only rely on the sample-rerank strategy (Zhao et al., 2022; Wu et al., 2023a), which is largely inefficient. Besides, the single-reference setting also biases the intrinsic evaluation, of keyphrase prediction models, as high-frequency topics in keyphrase labels may significantly outweigh the long-tail keyphrases.

To tackle these challenges, we propose *on-demand keyphrase generation*, a novel paradigm that predicts keyphrases conditioned on a *goal* phrase that specifies the high-level category or intent of the keyphrase (Figure 1). For existing keyphrase prediction models, this task is challenging as it requires the predictions to be not only capturing key information but also goal-conforming. Furthermore, the models are required to accept *open-vocabulary* goals, a significant step beyond predicting keyphrases with predefined categories or ontology (Park and Caragea, 2023).

To test on this new task, we meticulously curate and release METAKP, a large-scale on-demand keyphrase generation benchmark covering four datasets, 7500 documents, and 3760 unique goals from the news and the biomedical text domain. We build a scalable labeling pipeline that combines GPT-4 (OpenAI, 2023) and human annotators to construct high-quality goals from keyphrases (Figure 2). For evaluation, we design two tasks: judging the relevance of goals and generating goal-conforming keyphrases. For the latter, we employ the state-of-the-art evaluation method (Wu et al., 2023b) to conduct a semantic-based evaluation.

Using METAKP, we develop both supervised and unsupervised methods for on-demand keyphrase generation. For the supervised method, we design a multi-task fine-tuning approach to enable sequence-to-sequence pre-trained language models to self-determine the relevance of a goal and selectively generate keyphrases (Section 4.1). Then, in Section 4.2, we introduce an unsupervised self-consistency prompting approach leveraging the strong ability of large language models (LLMs) to propose goal-related keyphrase candidates and their propensity to predict high quality keyphrases with higher frequencies and ranks. Comprehensive experiments reveal the following insights:

1. METAKP represents a challenging benchmark for keyphrase generation. Flan-T5-XL, the strongest fine-tuned model, only achieves an average of 0.609 Satisfaction Rate across all the datasets, and zero-shot prompting GPT-4o, a strong LLM, only achieves 0.492 SR.
2. The proposed fine-tuning approach enables jointly learning goal relevance judgment and keyphrase generation without impeding each task's performance (Section 5.3).
3. The proposed self-consistency prompting approach greatly improves the performance of LLMs, enabling GPT-4o to achieve 0.548 SemF1, surpassing the performance of a fully fine-tuned BART-base model.
4. Supervised fine-tuning can fail to generalize on out-of-distribution testing data. By contrast, LLM-based unsupervised method achieves consistent performance in all the domains, especially in the news domain, where GPT-4o outperforms supervised Flan-T5-XL by 19% in out-of-distribution testing.

Finally, we demonstrate the potential of on-demand keyphrase generation as a general NLP infrastructure. Specifically, we use event detection for epidemics prediction (Parekh et al., 2024) as a test bed. By constructing simple goals from event ontology and attempting to extract relevant keyphrases from social media text, we show that an on-demand keyphrase generation model has the potential to extract epidemic-related trends similar to an event detection model trained on task-specific data. The benchmark and experimental code will be released to facilitate further research.

2 Related Work

Keyphrase Prediction with Types This work is closely related to prior work on modeling keyphrases with pre-defined types or categories. Early datasets are often derived from named entity recognition, where keyphrase spans are extracted with entity type tags (QasemiZadeh and Schumann, 2016; Augenstein et al., 2017; Luan et al., 2018). Notable modeling approaches include using intermediate task for training strong and transferable encoder representations (Park and Caragea, 2020) as well as multi-task fine-tuning (Park and Caragea, 2023). In addition, existing literature has explored inducing high-level type variable for more accurate keyphrase prediction, such as topic-guided keyphrase generation (Wang et al., 2019; Zhang et al., 2022a), hierarchical keyphrase generation (Wang et al., 2016; Chen et al., 2020; Zhang et al., 2022b), as well as keyphrase completion (Zhao et al., 2021). Compare to these prior work, our benchmark features a massive set of open-vocabulary goals with wide domain coverage. We design novel supervised and unsupervised modeling approaches that consider up-to-date techniques such as large language models.

On-Demand Information Extraction Our work resonates with the recent trend of designing flexible formulations for information extraction. For instance, Zhong et al. (2021) propose a query-focused formulation for the summarization task, and Zhang et al. (2023) further extend the task to

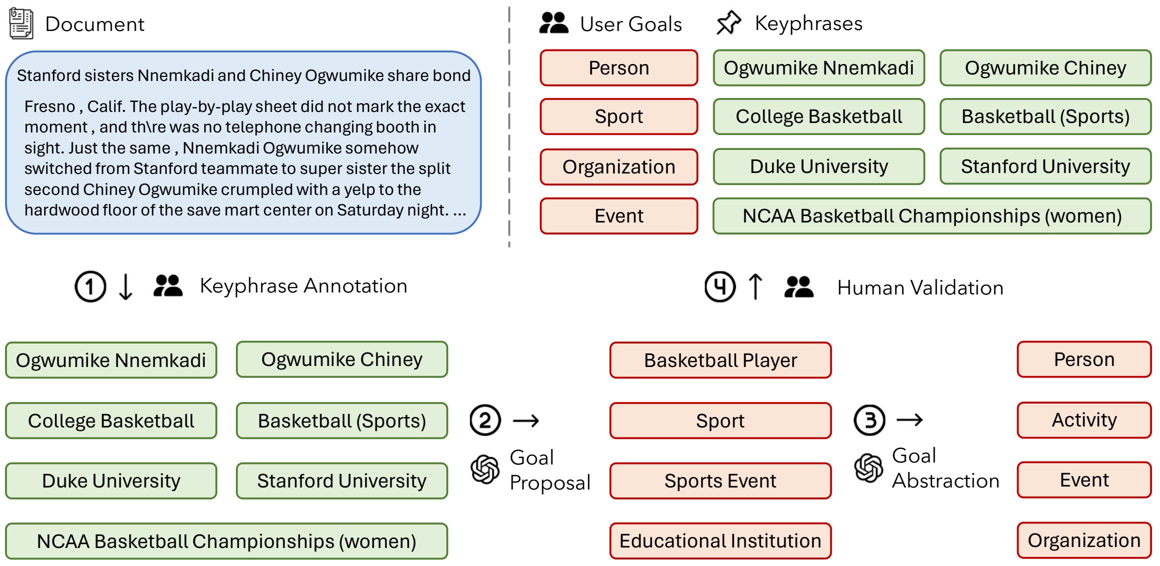

Figure 2: The annotation pipeline for METAKP. Starting from human-annotated keyphrases, GPT-4 is instructed to propose high-level goals and self-refine them. Finally, the goals are validated and filtered by humans.

include five constraints: Length, Extractiveness, Specificity, Topic, and Speaker. Recently, Jiao et al. (2023) introduce on-demand information extraction, where models are required to answer queries by extracting information from the associated text and organize it in a tabular format. By comparison, this work pioneers in defining and benchmarking the goal-following ability of keyphrase prediction models. Our resource and methodology lay the foundation for user-controllable keyphrase systems and flexible concept extraction infrastructures.

3 METAKP Benchmark

In this section, we formulate the on-demand keyphrase generation task and introduce the METAKP evaluation benchmark.

3.1 Problem Formulation

The traditional keyphrase prediction task is defined with a tuple: (document $\mathcal{X}$, reference set $\mathcal{Y}$). Given $\mathcal{X}$, a model directly generates all keyphrase hypotheses, with approximating $\mathcal{Y}$ as the goal. For on-demand keyphrase generation, we introduce an open-vocabulary goal phrase g which describes a category of keyphrases specified by the user. The target of the model, then, is to generate a set of keyphrases based on $(\mathcal{X}, g)$ to approximate the set of goal-conforming keyphrases $\mathcal{Y}_g \subseteq \mathcal{Y}$.

Figure 1 provides an intuitive example of the task. We note that for irrelevant goals, $\mathcal{Y}_g = \phi$, which means that an ideal model should not generate any keyphrases given such goals. In addition, although $\mathcal{Y}_g$ varies according to the goal, the universal set of keyphrases $\mathcal{Y}$ is assumed to be generally fixed. In other words, g could viewed as a query that specifies a target subset from $\mathcal{Y}$, which enables a wide range of choices for the modeling design.

3.2 Benchmark Creation Pipeline

To evaluate on-demand keyphrase generation, we curate METAKP, a large-scale multi-domain evaluation benchmark. The key challenge is to construct general, meaningful, and diverse goals that reflect high-level keyphrase types in real-world scenarios such as document indexing and search engines. To collect high quality goals, we design a model-in-the-loop annotation pipeline that combines GPT-4 (OpenAI, 2023) with human annotators to infer goals reversely from keyphrase annotations (Figure 2), with four steps detailed as follows.

Keyphrase Annotation by Human Given the document $\mathcal{X}$, human annotators specify the set of all the possible keyphrases $\mathcal{Y}$. For METAKP, we directly leverage the expert-curated keyphrases from the respective keyphrase prediction datasets.

Goal Proposal We instruct GPT-4 to propose a high-level goal for each of the keyphrases, and the same goal could be shared by multiple keyphrases[1]. Concretely, given $\mathcal{X}$, $\mathcal{Y}$, GPT-4 returns a mapping from goals to keyphrases. We present the prompt for this step in Appendix A.

Goal Abstraction After the previous step, a draft goal has been associated with each keyphrase. Although the proposed goals are relevant, we observe that they are sometimes overly specific. There-

[1]We use `gpt-4-0613` via the OpenAI API.

fore, we instruct GPT-4 to perform a round of *self-refinement*, where it attempts to propose a more abstract version for each of the goals in the previous round, or keep the original goals if they are already high-level enough. The full prompt for this step is presented in Appendix A.

Human Validation We qualitatively find that the outputs from two GPT-4 annotation iterations are sufficiently abstract and diverse. To further improve the quality of the goals and reduce the level of duplication, two of the authors conduct a round of filtering to obtain the final goal annotations. As this step does not entail adding new goals, the annotators achieve a high inter-annotator agreement (detailed in the next section) following the annotation guideline, which we present in Appendix A. Finally, we create an instance for each of the filtered goals, taking the form $(\mathcal{X}, g_i, \mathcal{Y}_{g_i})$.

3.3 Dataset Statistics

We execute the aforementioned goal construction pipeline on four keyphrase prediction datasets covering two domains: news and biomedical text. For each domain, we create both an in-distribution and an out-of-distribution test set.

- **KPTimes** (Gallina et al., 2019) is a large-scale keyphrase generation dataset in the news domain. The documents are sourced from from New York Times and the keyphrases are curated by professional editors.
- **DUC2001** (Wan and Xiao, 2008) is a widely used keypharse extraction dataset with news articles collected from TREC-9, paired with human-annotated keyphrases.
- **KPBiomed** (Houbre et al., 2022) is a large-scale dataset containing PubMed abstracts paired with keyphrases annotated by paper authors themselves.
- **Pubmed** (Schutz, 2008) is a traditional keyphrase extraction dataset in the biomedical domain with documents and keyphrases extracted from the PubMed Central.

We curate a test set using each of these datasets and construct two domain-specific train/validation sets sampled from the training sets from KPTimes and KPBiomed. Table 1 and Figure 3 presents the basic statistics of the final datasets. Besides its domain coverage, one strength of METAKP is its *diverse* coverage: together, the dataset covers 3760 unique goals, including diverse topics and subjects. While 40% of the instances correspond to the 10 most popular goals in each domain, a substantial number of goals also fall into the long tail distribution, posing significant new challenge in understanding the goal semantics.

Source	Split	#Doc	#Inst	#Goal	|Goal|	#KP/Goal
KPTimes	Train	1859	7502	1083	1.43	1.32
	Val	100	392	148	1.46	1.37
	Test	984	3836	679	1.41	1.33
DUC2001	Test	308	1642	549	1.50	1.53
KPBiomed	Train	1886	7807	1311	1.75	1.27
	Val	100	404	189	1.75	1.32
	Test	994	4136	865	1.76	1.27
Pubmed	Test	1269	4988	843	1.82	1.33

Table 1: Basic statistics of METAKP. #Inst = number of instances in the form $(\mathcal{X}, g, \mathcal{Y}_g)$. |Goal| refers to the average number of words in g. Finally, #KP/Goal corresponds to the average cardinality of $\mathcal{Y}_g$.

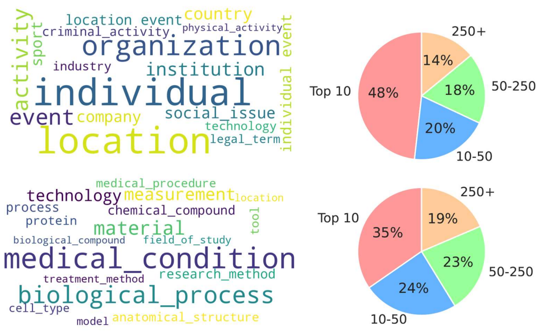

Figure 3: A visualization of the goal distribution for the news domain (top) and the biomedical domain (bottom). METAKP features both high-frequency goals and a diverse long-tail goal distribution.

To construct METAKP, the two-staged GPT-4 annotation costed approximately 500 USD, and the human annotators worked for approximately 80 hours in total on final data filtering. We randomly sample 50 documents each from KPTimes and KPBiomed, on which the annotators reach 0.699 Cohen's Kappa for inter-annotator agreement. Then, the annotators work on the rest documents separately. When ambiguous cases are found, a discussion is conducted to reach agreement.

Irrelevant Goal Sampling To test the ability of keyphrase generation models to abstain from generating keyphrases given irrelevant goals, for each document, we additionally construct a set of irrelevant goals. Concretely, we cluster the goals in the labelled data and use each document's existing goals as anchors to sample goals that are likely to be irrelevant to the document and thus it is unlikely that a keyphrase corresponding to the sampled goal exists for the document. We present the algorithm in the Appendix A.3. Using the algorithm, a balanced training set was created for training supervised methods for goal relevance judgment.

3.4 Evaluation Metric

With MetaKP, we design two tasks to comprehensively evaluate a model's ability to perform on-demand keyphrase generation.

Goal Relevance Assessment This task aims to test whether a model can correctly distinguish irrelevant goals that cannot yield any keyphrase from the relevant goals. As we will show in Section 6, this skill is also crucial to enable a wide application of on-demand keyphrase generation models. Following recent literature on abstention (Feng et al., 2024), we use **Abstain F1** as the evaluation metric, which is defined as the harmonic mean of the precision and the recall of a model refusing to generate keyphrases for irrelevant goals.

Goal-Oriented Keyphrase Generation Given document $\mathcal{X}$, a list of goals $g_1, g_2, ..., g_n$, and references $\mathcal{Y}_{g_1}, \mathcal{Y}_{g_2}, ..., \mathcal{Y}_{g_n}$, we evaluate a model's predictions $P_1, P_2, ..., P_n$ with two metrics:

1. **Reference Agreement**, which assesses the model's ability to generate keyphrases specifically corresponding to the goal g_i. Concretely, we calculate and report $SemF1(Y_{g_i}, P_i)$, following Wu et al. (2023b).
2. **Satisfactory Rate** (SR), which assesses the frequency of the model generating high-quality keyphrases. Concretely, we calculate and report $SR((\mathcal{Y}g_1, P_1), ..., (\mathcal{Y}g_n, P_n))$ as the percentage of goals that have $SemF1(Y_{g_i}, P_i)$ greater than a threshold[2].

4 Modeling Approach

In this section, we introduce two modeling approaches for on-demand keyphrase generation: a multi-task learning approach for fine-tuning sequence-to-sequence pre-trained language models, and a self-consistency decoding approach for prompting large language models (LLMs).

4.1 Multi-Task Supervised Fine-tuning

Previous literature has demonstrated the effectiveness of fine-tuning sequence-to-sequence pre-trained language models for keyphrase generation (Kulkarni et al., 2022; Wu et al., 2022, 2023a). However, it is unclear how these sequence prediction approaches could be adopted for on-demand keyphrase generation. To bridge this gap, we introduce a novel formulation to train a sequence-to-sequence model to autoregressively (1) assess the relevance of goals and (2) jointly consider the document as well as a desired goal to predict keyphrases.

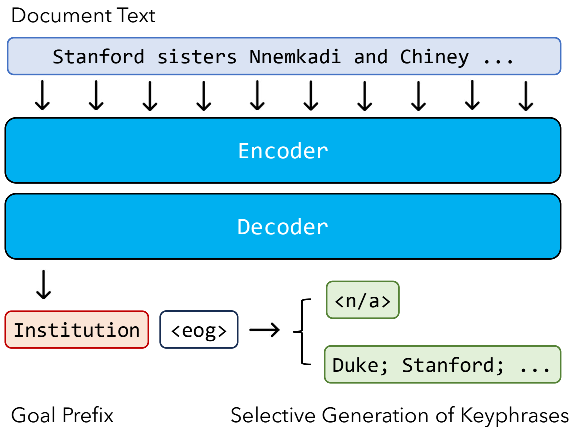

Figure 4: A visualization of the inference process of the proposed sequence-to-sequence generation approach. Based on the document and the goal prefix, the model self-decides the relevance of the goal and selectively generates the keyphrases for relevant goals only.

Concretely, we formulate on-demand keyphrase generation as a hierarchical composition of two token prediction tasks. As shown in Figure 4, with the document fed in the encoder, the decoder first models $P(g_i|\mathcal{X})$, the likelihood of g_i being a high-quality relevant goal proposed by real users. The model verbalizes this probability in $P(\text{<n/a>}|\mathcal{X}, g_i)$, a special token for rejecting irrelevant goals. If the goal is determined as relevant, the model proceeds generating the keyphrases according to the distribution $P(\mathcal{Y}_{g_i}|\mathcal{X}, g_i)$ it learned.

Inference We use prefix-controlled decoding for inference. g_i, followed by a special end-to-goal token <eog>, is fixed as the decoder's start of generation. Then, we use autoregressive decoding to let the model self-assess the relevance of goal and automatically decide the keyphrases to generate.

Training We design a multi-task learning procedure to directly supervise the model on $P(\text{<n/a>}|\mathcal{X}, g_i)$ and $P(\mathcal{Y}_{g_i}|\mathcal{X}, g_i)$ with a mixture of relevant and irrelevant goals. As the goals provided by users could be arbitrary, we do not directly supervise the model on $P(g_i|\mathcal{X})$.

[2]We fix $\tau = 0.6$. This decision is based Wu et al. (2023b), which suggests that the embedding model for $SemF1$ assigns a similarity score of approximately 0.6 for name variations.

Remark We note that the proposed approach has several advantages. First, both the goal relevance assessment and the keyphrase prediction process are streamlined in a single sequence prediction process, removing the need for separate architecture or inference pass. Second, since g_i is not fed to the encoder, our model avoids the goal being diluted by the long input context and enables efficient inference by reusing the encoded input representation for predicting keyphrases with different goals.

4.2 Prompting Large Language Models

Large language models (LLMs) that are tuned to follow human instructions have been shown to adapt well to a massive number of tasks defined through human queries (Ouyang et al., 2022; OpenAI, 2023). They have also been demonstrated to achieve promising keyphrase extraction or keyphrase generation performance, especially with semantic-based evaluation (Song et al., 2023; Wu et al., 2023b). As on-demand keyphrase extraction could be easily formulated as an instruction-following task, we investigate the potential of LLMs as an unsupervised approach. We start with a simple instruction for judging a goal's relevance:

```
Decide if you should reject the high-level
category given the title and abstract of a
document. One could use the high-level category
to write keyphrases from the document.
```

as well as another instruction for keyphrase generation based on a goal:

```
Generate present and absent keyphrases belonging
to the high-level category from the given text.
```

Our preliminary experiments show that the first instruction already achieves a strong performance in deciding the goal relevance, even approaching supervised models (Section 5.2). However, when it comes to keyphrase generation, LLMs intriguingly misinterpret the task as named entity extraction: they often generate an almost exhaustive list of goal-related entities. To correct this behavior, we hypothesize that LLMs tend to generate salient entities more frequently and at an earlier location of the prediction sequence. Inspired by Wang et al. (2023), we thus design a novel self-consistency decoding process to leverage the rank and frequency information in LLMs' samples to filter out phrases that encode the most important information.

Concretely, using the same instruction and input, we sample K prediction sequences $(s_1, ..., s_K)$ from the LLM independently, each of which contains a variable number of keyphrases. Then, for each keyphrase p, we define its saliency score as:

$$score(p) = \frac{freq(p)}{K} \times \frac{freq(p)}{\sum_{i=1,...,K} rank(s_i,p)},$$

where $freq(p)$ returns the frequency of p in all the samples and $rank(s_i, p)$ returns the rank of p in s_i (starting from 1) or 0 if $p \notin s_i$. The first term rewards keyphrases that frequently present in the samples, and the second term rewards keyphrases with a higher rank. Together, the score is defined to range 0 from 1 regardless of the number of samples or the number of keyphrases a model generates per sample. Finally, we apply threshold filtering and only retain the high quality keyphrases with $score(p)$ greater than or equal to a threshold τ.

5 Experiments

5.1 Experimental Setup

Supervised Fine-tuning Using the proposed objective, we fine-tune four sequence-to-sequence models: `BART-base/large` (Lewis et al., 2020) and `Flan-T5-large/XL` (Longpre et al., 2023), with diverse sizes ranging from 140M to 3B. We train the models for 20 epochs with batch size 64, learning rate 3e-5, the Adam optimizer, and a linear decay with 50 warmup steps. The best model checkpoint is chosen based on the keyphrase generation performance on the validation set.

Prompting We use `gpt-3.5-turbo-0125` and the `gpt-4o-2024-05-13` models via the OpenAI API. We will denote the models as `GPT-3.5-Turbo` and `GPT-4o`. We use separate prompts for goal relevance judgment and on-demand keyphrase generation. For the first task, greedy search is used. For the second task, we generate 10 samples with p = 0.95 and temperature = 0.7. The output length is limited 30 tokens, which can accommodate approximately 10 keyphrases. Finally, for filtering, we use $\tau = 0.3$ for all the datasets.

We document the full implementation details in Appendix B, including the prompt for language language models, the post-processing process, as well as the details for hyperparameter tuning.

5.2 Main Results

We present the main results for the two tasks in Figure 5 and Table 2.

Goal Relevance Assessment According to Figure 5, we find both supervised fine-tuning and unsupervised prompting reaches a high performance for

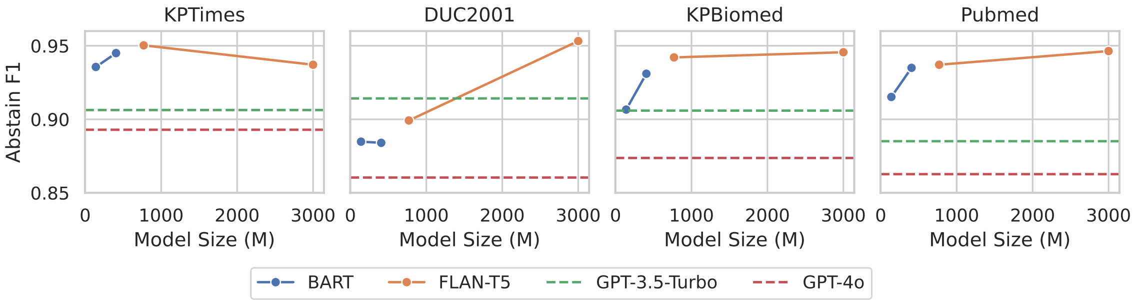

Figure 5: Goal relevance judgment results of different types of models. Zero-shot prompting LLMs achieves a high performance, despite slightly falling below supervised models. Also, GPT-4o does improve over GPT-3.5-Turbo.

Model	Size	Method	KPTimes✣		DUC2001✣		KPBiomed✿		Pubmed✿		Average	
			SemF1	SR	SemF1	SR	SemF1	SR	SemF1	SR	SemF1	SR
Supervised Methods												
`BART-base`	140M	No Goal	0.395	0.192	0.299	0.089	0.300	0.107	0.305	0.196	0.325	0.146
		MetaKP	0.728	0.699	0.447	0.319	0.508	0.417	0.504	0.406	0.547	0.460
`BART-large`	406M	No Goal	0.399	0.196	0.306	0.081	0.297	0.074	0.290	0.070	0.323	0.105
		MetaKP	0.752	0.738	0.469	0.336	0.545	0.461	0.529	0.437	0.574	0.493
`Flan-T5-large`	770M	MetaKP	**0.765**	**0.758**	0.488	0.360	0.578	0.506	0.572	0.501	0.601	0.531
`Flan-T5-XL`	3B	MetaKP	0.763	0.757	0.484	0.361	**0.594**†	**0.530**†	**0.593**†	**0.526**†	**0.609**†	**0.544**†
Unsupervised Methods												
`GPT-3.5-Turbo`	-	Zero-Shot	0.452	0.221	0.499	0.290	0.421	0.166	0.444	0.217	0.454	0.224
		Sample + SC	0.518	0.406	0.572	0.516	0.513	0.423	0.472	0.376	0.519	0.430
`GPT-4o`	-	Zero-Shot	0.491	0.281	0.526	0.374	0.480	0.278	0.469	0.262	0.492	0.299
		Sample + SC	0.552	0.460	**0.578**†	**0.535**†	0.529	0.451	0.532	0.453	0.548	0.475

Table 2: Experiment results of supervised and unsupervised methods on-demand keyphrase generation. We use different superscripts to denote results that are reported using the models trained on KPTimes (✣) and KPBiomed (✿). SR = satisfaction rate. SC = self-consistency prompting The best results are boldfaced. †statistically significantly better than the second highest result with $p < 0.01$, tested via paired t-test.

assessing whether a goal, as indicated by over 0.85 Abstain F1 scores across all datasets. As model size scales, the out-of-distribution performance scales more readily, while the in-distribution performance plateaus at `Flan-T5-large`. With large language models, we observe strong performance especially on DUC2001, surpassing the performance of `Flan-T5-large` trained on KPTimes.

Keyphrase Generation The main results for keyphrase generation are presented in Table 2. For supervised methods, we additionally include a "No Goal" baseline, where the model is fine-tuned to generate all the keyphrases for the same document at once. For both `BART-base` and `BART-large`, this baseline achieves a low performance, indicating the challenging nature of directly leveraging a keyphrase generation model for the proposed task. By comparison, the proposed goal-directed fine-tuning approach improves the performance by a large margin, with the best `Flan-T5-XL` model achieving 0.609 SemF1 and 0.544 satisfaction rate. On the other hand, directly zero-shot prompting large language models already achieves more superior performance compared to the supervised models trained without any goal. The proposed self-consistency further improves the performance substantially, allowing `GPT-4o` achieve 0.548 SemF1 and 0.475 satisfaction rate. Notably, results demonstrate that the LLM-based approach has the potential to be more generalizable. On DUC2001, all supervised models trained on KPTimes demonstrate a poor performance. By contrast, both `GPT-3.5-Turbo` and `GPT-4o` are able to surpass the performance of all supervised models.

5.3 Analyses

Which parameter affects LLMs the most? In Figure 6, we use use KPTimes' validation set to investigate the sensitivity of the LLM-based approach to three hyperparameters: number of samples (K), threshold τ, and context length of the input. Although multiple samples are essential to high performance, more samples after two only help marginally. In addition, our method is insensitive to the threshold setting - the best performance can be obtained by multiple settings between 0.25 and 0.45. Finally, while `GPT-3.5-Turbo` exhibits a slight performance drop with longer context, `GPT-4o` is robust to context length variations.

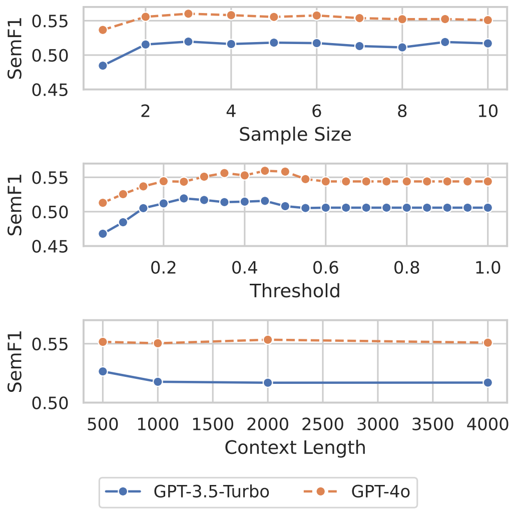

Figure 6: Sensitivity of the self-consistency prompting approach's performance to number of samples, settings of threshold τ, and the input length on KPTimes. The results on KPBiomed is presented in Figure 12.

Objective	ID AF1	ID SR	OOD AF1	OOD SR
Training on KPTimes				
Multi-task Learning	**0.936**	**0.699**	0.885	**0.319**
Goal Relevance Only	0.928	-	**0.898**	-
Keyphrase Only	-	0.692	-	0.316
Training on KPBiomed				
Multi-task Learning	0.907	0.417	0.915	0.406
Goal Relevance Only	**0.917**	-	**0.916**	-
Keyphrase Only	-	**0.425**	-	**0.407**

Table 3: Ablation study on the multi-task learning setup. AF1 = Abstain F1, SR = Satisfaction Rate.

Does multi-task learning harm each individual task's performance? In Table 3, we conduct an ablation study with `BART-base` on the supervised training loss. For each ablated component, we mask out the corresponding tokens when calculating the loss. Overall, combining the two learning objectives do not significantly harm the performance compared to only learning individual tasks, while incurring much less computational overhead. In fact, on KPTimes, the two tasks are constructive - learning goal relevance helps generating better goal-conforming keyphrases, and vice versa.

6 METAKP in the Wild: Event Detection

Finally, we demonstrate the potential of on-demand keyphrase generation as general NLP infrastructure, using event detection (ED) as a case study.

We leverage the testing dataset used in SPEED (Parekh et al., 2024), which contains time-stamped social media posts related to Monkeypox[3]. From

[3]We solicited the dataset and outputs from the authors.

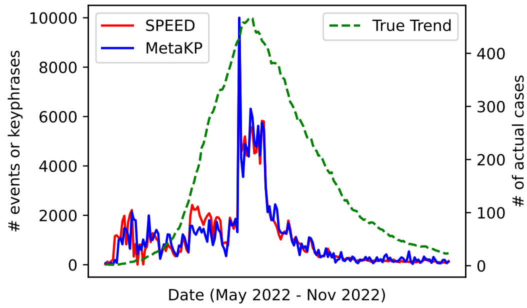

Figure 7: Number of events/keyphrases extracted for Monkeypox as a function of time. The true trend and SPEED outputs are solicited from Parekh et al. (2024).

the SPEED ontology, we curate seven epidemic-related goals: *disease infection, epidemic spread, epidemic prevention, epidemic control, symptom, recover from disease, death from epidemic*. Then, we run a `FLAN-T5-large` model trained on all training data from METAKP to assess the relevance of each goal against each social media post. If the probability of `<n/a>` following `<eog>` is greater than 0.001, the goal is judged relevant to the post and thus the underlying event is likely.

As shown in Figure 7, we observe that this keyphrase-based paradigm is able to extract trends that are similar to an ED model trained on SPEED (Parekh et al., 2024). Intuitively, given a sentence containing "getting vaccination", instead of focusing on the trigger "get", on-demand keyphrase generation is able focus more on "vaccination", given the goal "epidemic control". In this way, on-demand keyphrase generation models can both be naturally repurposed for ED and also promises to extract supporting topics related to the the event.

7 Conclusion

We introduce on-demand keyphrase generation, targeting the need for dynamic, goal-oriented keyphrase prediction tailored to diverse applications and user demands. A large-scale, multi-domain, human-verified benchmark METAKP was curated and introduced. We designed and evaluated both supervised and unsupervised methods on METAKP, highlighting the strengths of self-consistency prompting with large language models. This approach significantly outperformed traditional fine-tuning methods under domain shifts, showcasing its robustness and the broader applicability of our methodology. Finally, we underscore the versatility of on-demand keyphrase generation in practical applications such as epidemic event extraction, promising a new direction for keyphrase generation as general NLP infrastructure.

Limitations

In this work, we propose the novel on-demand keyphrase generation paradigm. In the future, several exciting directions exist for extending the paradigm as well as the METAKP benchmark:

1. **Multi-lingual Keyphrase Generation**. METAKP only covers data in English. Further benchmarking and enhancing the multilingual and cross-lingual on-demand keyphrase generation ability is an important future direction.
2. **Wider Domain Coverage**. We mainly focus on the news and the biomedical text domain as they have been shown as important application domains for keyphrase generation.
3. **Flexible Instructions**. In this work, the "demand" from the users are generally defined as topics or categories of keyphrases. However, future work could expand this definition to include demands that specify stylistic constraints such as the number of keyphrases, the length, and their formality.

Ethics Statement

As a new task and paradigm, on-demand keyphrase generation may bring new security risks and ethical concerns. To begin with, although keyphrase generation models generally have outstanding understanding of phrase saliency, they generally have a shallower understanding of semantics and factuality. Thus, when pairing keyphrases with goals, potential misinformation could be created. For instance, when queried with "cure" as a goal, a model may return certain concepts that are factually wrong. In addition, when queries contain certain occupations as goals, a keyphrase generation model may reinforce existing gender stereotypes by selectively generating and ignoring entities with a certain gender. We view these possibilities as potential risks and encourage a thorough redteaming process before deploying on-demand keyphrase generation systems in real-world products.

We use KPTimes and KPBiomed data distributed by the original authors. For DUC2001 and PubMed, we access the data via ake-datasets[4]. KPTimes was released under Apache-2.0 license, and we cannot find licensing information for DUC2001, KPBiomed, and PubMed. ake-datasets was also released under Apache-2.0. No additional preprocessing is performed in METAKP except lowercasing and tokenization. While we mainly rely on the original authors for dataset screening to remove sensitive and harmful information, we also actively monitor the data quality during in the human filtering process and remove any document that could cause privacy or ethics concerns. As OpenAI models are involved in the data curation process, our code and datasets will be released with MIT license with a research-only use permission.

[4] https://github.com/boudinfl/ake-datasets

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Gábor Berend. 2011. Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

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, Online. Association for Computational Linguistics.

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.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-llm collaboration. *CoRR*, abs/2402.00367.

Ygor Gallina, Florian Boudin, and Beatrice Daille. 2019. KPTimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135, Tokyo, Japan. Association for Computational Linguistics.

Maël Houbre, Florian Boudin, and Beatrice Daille. 2022. A large-scale dataset for biomedical keyphrase generation. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 47–53, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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.

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. 2023. Instruct and extract: Instruction tuning for on-demand information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore. Association for Computational Linguistics.

Youngsam Kim, Munhyong Kim, Andrew Cattle, Julia Otmakhova, Suzi Park, and Hyopil Shin. 2013. Applying graph-based keyword extraction to document retrieval. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 864–868, Nagoya, Japan. Asian Federation of Natural Language Processing.

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.

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.

Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael R. Lyu, and Irwin King. 2020. Unsupervised text generation by learning from search. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. 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.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Tanmay Parekh, Anh Mac, Jiarui Yu, Yuxuan Dong, Syed Shahriar, Bonnie Liu, Eric Yang, Kuan-Hao Huang, Wei Wang, Nanyun Peng, and Kai-Wei Chang. 2024. Event detection from social media for epidemic prediction. *Preprint*, arXiv:2404.01679.

Seo Park and Cornelia Caragea. 2023. Multi-task knowledge distillation with embedding constraints for scholarly keyphrase boundary classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13026–13042, Singapore. Association for Computational Linguistics.

Seoyeon Park and Cornelia Caragea. 2020. Scientific keyphrase identification and classification by pre-trained language models intermediate task transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5409–5419, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Behrang QasemiZadeh and Anne-Kathrin Schumann. 2016. The ACL RD-TEC 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1862–1868, Portorož, Slovenia. European Language Resources Association (ELRA).

Alexander Schutz. 2008. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods.

Mingyang Song, Haiyun Jiang, Shuming Shi, Songfang Yao, Shilong Lu, Yi Feng, Huafeng Liu, and Liping Jing. 2023. Is chatgpt a good keyphrase generator? a preliminary study. *arXiv preprint arXiv:2303.13001*.

Yixuan Tang, Weilong Huang, Qi Liu, Anthony K. H. Tung, Xiaoli Wang, Jisong Yang, and Beibei Zhang. 2017. Qalink: Enriching text documents with relevant q&a site contents. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019. Topic-aware neural keyphrase generation for social media language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2516–2526, Florence, Italy. Association for Computational Linguistics.

Yunli Wang, Yong Jin, Xiaodan Zhu, and Cyril Goutte. 2016. Extracting discriminative keyphrases with learned semantic hierarchies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 932–942, Osaka, Japan. The COLING 2016 Organizing Committee.

Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255.

Di Wu, Wasi Ahmad, and Kai-Wei Chang. 2023a. 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 Ahmad, Sunipa Dev, and Kai-Wei Chang. 2022. Representation learning for resource-constrained keyphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 700–716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Di Wu, Da Yin, and Kai-Wei Chang. 2023b. Kpeval: Towards fine-grained semantic-based evaluation of keyphrase extraction and generation systems.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. 2023. MACSum: Controllable summarization with mixed attributes. *Transactions of the Association for Computational Linguistics*, 11:787–803.

Yuxiang Zhang, Tao Jiang, Tianyu Yang, Xiaoli Li, and Suge Wang. 2022a. HTKG: deep keyphrase generation with neural hierarchical topic guidance. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1044–1054. ACM.

Yuxiang Zhang, Tianyu Yang, Tao Jiang, Xiaoli Li, and Suge Wang. 2022b. Hyperbolic deep keyphrase generation. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2022, Grenoble, France, September 19-23, 2022, Proceedings, Part II*, volume 13714 of *Lecture Notes in Computer Science*, 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.

Yu Zhao, Jia Song, Huali Feng, Fuzhen Zhuang, Qing Li, Xiaojie Wang, and Ji Liu. 2021. Deep keyphrase completion. *CoRR*, abs/2111.01910.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Supplementary Material: Appendices

A METAKP Construction Details

In this section, we describe the details of the construction process of METAKP.

A.1 GPT-4 Annotation

Goal Proposal In Figure 8, we show the prompt used to instruct GPT-4 to propose goals from the document and human-annotated keyphrases. We truncate the document body to four sentences as its role is only providing essential contextual. The LLM is instructed to propose all the goals for all the keyphrases together, which helps the model group together keyphrases that share the same goal.

```
Document Title: {title}
First 4 sentences of the document body: {body}

Keyphrases (separated by ";"): {keyphrases}

For each keyphrase, generate an abstract category for the keyphrase. Examples include process, task, material, tool, measurement, model, technology, and metric etc. Do not limit yourself to the examples. Make sure that the categories are informative in the domain of science and appearing natural as if that assigned by a well-read user. Return a list of dictionaries, each with two keys - "keyphrase" and "category". If two keyphrases have the same category, make sure that they are labelled with the same phrase. Do not change how the keyphrases appear, including their cases. Return json only and do not say anything else.
```

Figure 8: Prompt used for instructing GPT-4 to generate the goals from a document and keyphrases.

Goal Refinement Then, we instruct GPT-4 to refine the goals by trying to generate more abstract versions of them. The prompt is shown in Figure 9. As we perform the refinement directly from the chat history of the previous step, we omit the previous prompt and step 1 model outputs.

```
... step 1 prompt and model outputs ...

Can you make the categories more abstract, yet still informative to the keyphrase? If the categories are already abstract enough, you do not need to change. Return json only.
```

Figure 9: Prompt used for instructing GPT-4 to improve the abstractiveness of the proposed goals.

For both of the steps, we use greedy search and cap the output to 400 tokens. We parse the results string into json format to extract the goals.

A.2 Human Validation

Next, based on the two rounds of proposed goals, the two authors (student researchers familiar with NLP and the keyphrase generation task) filter out high quality goals as the final benchmarking dataset. We emphasize that this decision is required due to the nature of the task, which requires expert annotators to ensure a high data quality. The consent to use and release the annotation traces was obtained from both of the authors. The type of research conducted by this work is automatically determined exempt from by the authors' institution's ethics review board. We design and enforce two major guidelines during the annotation process:

1. Remove a goal if it is semantically equivalent to or a subtype of some another goal that is more abstract.
2. Remove a goal if it so abstract that it could also enclose other keyphrases not currently paired with the goal. This criterion includes overly vague goals (e.g., "concept") and goals that corresponds to the topic of the entire pssage (e.g., "chemistry concepts").

As mentioned in Section 3, this process allows the annotator reach a high inter-annotator agreement of 0.699 Cohen's Kappa. In addition, the annotator actively engage in a discussion whenever ambiguous cases are found. Finally, we conduct a rule-based postprocessing with two stages.

1. **Goal Removal**. We remove the following goals as they represent overly general goals: `entity, process, concept`.
2. **Goal Unification**. We merge the following goal labels as they represent the same meaning. Table 4 presents the source and target goals. Note that to preserve the diversity of the goals, we refrain from merging aggressively and only merge the basic cases that may be result from annotation discrepancy.

A.3 Negative sampling Algorithm

To construct the training and evaluation data for evaluating the model's ability to reject irrelevant goals, we design a simple algorithm to sample irrelevant goals. Concretely, we pool together all the existing goals from the same dataset as the universal goal set and leverage the phrase embedding model released by (Wu et al., 2023b) to embed all the phrases. Then, for each goal from the docu-

Source Goals	Target Goals
place, geographical location	location
person, people, individual person	individual
geopolitical entity	country
... event	event
profession	occupation
belief system	religion
incident outcome	outcome
subject	topic
incident	event
... equipment	equipment
... procedure	procedure

Table 4: Goal merging directions for METAKP label cleaning. We replace all occurrences of source goals with target goals.

ment, we use it as an anchor to retrieve $d\%$ most dissimilar goals. We use $d = 50$ for all the datasets. From these goals, we sample a goal that is not associated with the document as the irrelevant goal according to the frequency distribution of these goals appearing as relevant goals in the final dataset. We additionally design a frequency match constraint, which enforces that the frequency of a goal g appearing as an irrelevant goal should not exceed the frequency it appears as a relevant goal. In practice, the frequency match constraint is applied first. If no eligible goals remain, we sample a goal from the $d\%$ most dissimilar goals according to frequency.

B Implementation Details

B.1 Supervised Fine-tuning

For multi-task learning with `BART` and `Flan-T5`, we base our implementation on the Huggingface Transformers implementations provided by (Wu et al., 2023a) and train for 20 epochs with early stopping. We use learning rate 3e-5, linear decay, batch size 64, and the AdamW optimizer. Due to the context limitations of `Flan-T5`, all the input documents for `BART` and `Flan-T5` are truncated to 512 tokens to enable a fair comparison. We perform a careful hyperparameter search over the learning rate, batch size, and warm-up steps. The corresponding search spaces are {1e-5, 3e-5, 6e-5, 1e-4}, {16, 32, 64, 128}, and {50, 100, 250, 500}. The best hyperparameters are chosen based on the performance on the validation set. To decode from the fine-tuned models, we fix the decoder's prefix using the constrained decoding functionalities provided by Huggingface Transformers and use greedy search to complete the suffix.

The fine-tuning experiments are performed on a local GPU server with eight Nvidia RTX A6000 GPUs (48G each). We use gradient accumulation to achieve the desired batch sizes. Fine-tuning `BART-base`, `BART-base`, `Flan-T5-large`, and `Flan-T5-XL` take, respectively.

B.2 Large Language Models

We present the prompts for prompting large language models for goal relevance judgment and goal-conforming keyphrase generation in Figure 10 and Figure 11.

```
In this task you will need to decide if you
should reject the high-level category given the
title and abstract of a document.  One could
use the high-level category to write keyphrases
from the document. If you decide the category
is relevant to the document, generate yes; if
the category is not relevant, generate no. Do
not output anything else.

Document Title: {title}
Document Abstract: {body}

High-level Category: {goal}
Relevant? (yes or no):
```

Figure 10: Prompt used for goal relevance judgment.

```
Generate  present  and  absent  keyphrases
belonging to the high-level category from the
given text, separated by commas. Do not output
anything else.

Document Title: {title}
Document Abstract: {body}

High-level Category: {goal}
Keyphrases (Must be of category "{goal}"):
```

Figure 11: Prompt used for on-demand keyphrase generation with LLMs.

For all the results reported in the paper, we use `gpt-3.5-turbo-0125` and the `gpt-4o-2024-05-13` models via the OpenAI API.

For goal relevance judgment, we use greedy decoding and record the yes/no predictions for evaluation. The document body is truncated to the first five sentences as we find providing longer context barely improves the performance.

For on-demand keyphrase generation, the input length is truncated to 4000 tokens. We generate 10 samples with p = 0.95 and temperature = 0.7. The output length is limited to 30 tokens, which accommodate approximately 10 keyphrases. Finally, for filtering, we set a fixed threshold $\tau = 0.3$. We lower-case all the outputs and use a string matching algorithm to remove excessive parts generated by the model such as "present keyphrases: ". The method's sensitivity to the hyperparameter settings is presented in Figure 6 and Figure 12.

Since the proposed LLM-based methods are unsupervised, we refrain from extensively tuning the hyperparameters. The only exception is that we use the validation sets to determine a reasonable good setting of the sample size K and the threshold τ, which is uniformly applied to all the datasets.

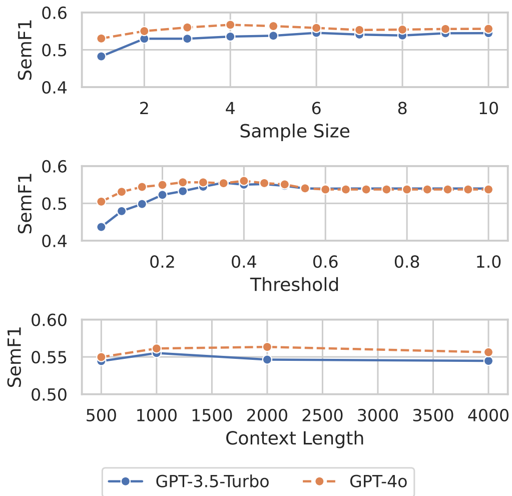

Figure 12: Sensitivity of the self-consistency prompting approach's performance to number of samples, settings of threshold τ, and the input length on KPBiomed.

C Qualitative Study

In Figure 13 and Figure 14, we present and compare the outputs of `Flan-T5-XL`, zero-shot sampling from `GPT-4o`, and self-consistency sampling from `GPT-4o` in two domains. Compared to supervised models, which often generates suboptimal keyphrases under distribution shift, GPT-4o exhibits consistent high recall across domains, and the self-consistency reranking process further filters high quality goals from the zero-shot keyphrase predictions across multiple samples.

Domain: News

Source: KPTimes

Title: more connecticut schools fall short of “no child left behind" standards

Abstract: since [digit] , when president bush signed the federal no child left behind law , schools in connecticut have scrambled to revamp curriculums , step up professional development for teachers and continually assess students ’ test scores to comply with the law ’ s requirements . it is an effort that has dominated the agendas of school officials not just in connecticut , but all over the nation , and not everyone is happy that test results have become such a focus . when results of the latest test scores were announced a few weeks ago , about [digit] percent , or [digit] , of the public schools in connecticut did not make the grade under the federal law , state officials said . the state added [digit] schools to its list of schools that failed to meet the federal benchmarks .

Goal 1: institution
Reference: "education and schools"
Prediction (`Flan-T5-XL`): "education and schools"
Prediction (`GPT-4o (zero-shot)`): "connecticut schools", "federal government", "state department of education", "norwalk public schools", "greenwich school district", "greenwich public schools"
Prediction (`GPT-4o (self-consistency)`): "connecticut schools", "state department of education"

Goal 2: law
Reference: "no child left behind act"
Prediction (`Flan-T5-XL`): "no child left behind act"
Prediction (`GPT-4o (zero-shot)`): "no child left behind", "federal no child left behind law", "federal law", "federal government", "new accountability system", "adequate yearly progress"
Prediction (`GPT-4o (self-consistency)`): "no child left behind", "no child left behind law"

Source: DUC2001 (out-of-distribution)

Title: (empty)

Abstract: millions of gallons of crude oil that spilled when a tanker ran aground spread across a wildlife - rich stretch of ocean saturday , and alaska ’ s chief environmental officer criticized cleanup efforts as too slow . the biggest oil spill in u . s . history created a slick about seven miles long and seven miles wide in prince william sound . the coast guard said only reef island and the western edge of bligh island had been touched by the slick . “ this situation , i think , was everyone ’ s secret nightmare about what could happen with oil traffic in the sound ,” said dennis kelso , commissioner of the alaska department of environmental conservation .

Goal 1: substance
Reference: "crude oil"
Prediction (`Flan-T5-XL`): "oil (petroleum) and gasoline"
Prediction (`GPT-4o (zero-shot)`): "crude oil", "oil spill", "oil pollution", "north slope crude oil", "spilled oil", "leaking oil", "oil slick", "spilled crude oil"
Prediction (`GPT-4o (self-consistency)`): "crude oil"

Goal 2: action
Reference: "cleanup efforts"
Prediction (`Flan-T5-XL`): "accidents and safety"
Prediction (`GPT-4o (zero-shot)`): "criticized cleanup efforts", "created a slick", "ran hard aground", "halted early", "begin pumping", "removing oil", "placed a boom"
Prediction (`GPT-4o (self-consistency)`): "spread across", "criticized cleanup efforts"

Source: DUC2001 (out-of-distribution)

Title: (empty)

Abstract: the clinton administration will soon announce support for a north american development bank , which would fund projects in communities hit by job losses resulting from the north american free trade agreement . the so - called nadbank has been strongly supported by congressman esteban torres , who has insisted on some sort of lending institution to support adjustment throughout the continent . agreement by the administration is expected to bring mr torres and at least [digit] other hispanic congressmen into the pro - nafta fold . the administration believes it can garner [digit] - [digit] pro - nafta votes , out of the [digit] needed .

Goal 1: economic issue
Reference: "job losses"
Prediction (`Flan-T5-XL`): "jobs"
Prediction (`GPT-4o (zero-shot)`): "north american development bank", "job losses", "north american free trade agreement", "lending institution", "pro-nafta votes", "anti-nafta public opinion"
Prediction (`GPT-4o (self-consistency)`): "north american development bank", "clinton administration", "job losses""

Goal 2: political entity
Reference: "clinton administration"
Prediction (`Flan-T5-XL`): "united states politics and government"
Prediction (`GPT-4o (zero-shot)`): "clinton administration", "congressman esteban torres", "hispanic congressmen", "white house", "president bill clinton"
Prediction (`GPT-4o (self-consistency)`): "clinton administration", "north american development bank"

Figure 13: Examples of on-demand keyphrase generation instances and model outputs in the news domain.

Domain: Biomedical Text

Source: KPBiomed

Title: contemporary trend of acute kidney injury incidence and incremental costs among us patients undergoing percutaneous coronary procedures .

Abstract: objectives to assess national trends of acute kidney injury (aki) incidence , incremental costs , risk factors , and readmissions among patients undergoing coronary angiography (cag) and / or percutaneous coronary intervention (pci) during [digit] - [digit] . background aki remains a serious complication for patients undergoing cag / pci . evidence is lacking in contemporary aki trends and its impact on hospital resource utilization . methods patients who underwent cag / pci procedures in [digit] hospitals were identified from premier healthcare database . aki was defined by icd - [digit] / [digit] diagnosis codes ([digit] .

Goal 1: medical condition

Reference: "acute kidney injury", "chronic kidney disease", "nephropathy"

Prediction (`Flan-T5-XL`): "acute kidney injury"

Prediction (`GPT-4o (zero-shot)`): "acute kidney injury", "chronic kidney disease", "anemia", "diabetes"

Prediction (`GPT-4o (self-consistency)`): "acute kidney injury", "chronic kidney disease", "anemia"

Goal 2: medical procedure

Reference: "percutaneous coronary intervention"

Prediction (`Flan-T5-XL`): "percutaneous coronary intervention"

Prediction (`GPT-4o (zero-shot)`): "percutaneous coronary intervention", "coronary angiography", "coronary procedures", "inpatient procedures", "outpatient procedure"

Prediction (`GPT-4o (self-consistency)`): "percutaneous coronary intervention", "coronary angiography"

Source: PubMed (out-of-distribution)

Title: surviving sepsis campaign : international guidelines for management of severe sepsis and septic shock : [digit]

Abstract: objective to provide an update to the original surviving sepsis campaign clinical management guidelines , Ž01c surviving sepsis campaign guidelines for management of severe sepsis and septic shock ,Ž01d published in [digit] . introduction severe sepsis (acute organ dysfunction secondary to infection) and septic shock (severe sepsis plus hypotension not reversed with fluid resuscitation) are major healthcare problems , affecting millions of individuals around the world each year , killing one in four (and often more), and increasing in incidence [[digit] Ž013 [digit]]. similar to polytrauma , acute myocardial infarction , or stroke , the speed and appropriateness of therapy administered in the initial hours after severe sepsis develops are likely to influence outcome .

Goal 1: medical condition

Reference: "sepsis", "severe sepsis", "septic shock", "sepsis syndrome", "infection"

Prediction (`Flan-T5-XL`): "sepsis"

Prediction (`GPT-4o (zero-shot)`): "acute kidney injury", "chronic kidney disease", "anemia", "diabetes"

Prediction (`GPT-4o (self-consistency)`): "severe sepsis", "septic shock"

Goal 2: healthcare initiative

Reference: "surviving sepsis campaign"

Prediction (`Flan-T5-XL`): "surviving sepsis campaign"

Prediction (`GPT-4o (zero-shot)`): "surviving sepsis campaign", "international guidelines", "management of severe sepsis", "septic shock", "clinical management guidelines", "evidence-based methodology"

Prediction (`GPT-4o (self-consistency)`): "surviving sepsis campaign"

Source: PubMed (out-of-distribution)

Title: keratinocyte serum - free medium maintains long - term liver gene expression and function in cultured rat hepatocytes by preventing the loss of liver - enriched transcription factors

Abstract: freshly isolated hepatocytes rapidly lose their differentiated properties when placed in culture . therefore , production of a simple culture system for maintaining the phenotype of hepatocytes in culture would greatly facilitate their study . our aim was to identify conditions that could maintain the differentiated properties of hepatocytes for up to [digit] days of culture . adult rat hepatocytes were isolated and attached in williams Ž019 medium e containing [digit] % serum . the medium was changed to either fresh williams Ž019 medium e or keratinocyte serum - free medium supplemented with dexamethasone , epidermal growth factor and pituitary gland extract .

Goal 1: biological extract

Reference: "pituitary gland extract"

Prediction (`Flan-T5-XL`): "pituitary gland extract"

Prediction (`GPT-4o (zero-shot)`): "keratinocyte serum-free medium", "WilliamsŽ019 medium E", "dexamethasone", "epidermal growth factor", "pituitary gland extract"

Prediction (`GPT-4o (self-consistency)`): "keratinocyte serum-free medium", "keratinocyte serum"

Goal 2: molecular biology technique

Reference: "reverse transcription polymerase chain reaction"

Prediction (`Flan-T5-XL`): "cell culture"

Prediction (`GPT-4o (zero-shot)`): "immunohistochemistry", "western blotting", "rt-pcr", "immunofluorescence staining", "collagenase perfusion technique"

Prediction (`GPT-4o (self-consistency)`): "western blotting", "rt-pcr", "immunohistochemistry"

Figure 14: Examples of on-demand keyphrase generation instances and model outputs in the biomedical domain.