KPEVAL: Towards Fine-Grained Semantic-Based Keyphrase Evaluation

Di Wu, Da Yin, Kai-Wei Chang
University of California, Los Angeles
{diwu,da.yin,kwchang}@cs.ucla.edu

Abstract

Despite the significant advancements in keyphrase extraction and keyphrase generation methods, the predominant approach for evaluation mainly relies on exact matching with human references. This scheme fails to recognize systems that generate keyphrases semantically equivalent to the references or diverse keyphrases that carry practical utility. To better assess the capability of keyphrase systems, we propose KPEVAL, a comprehensive evaluation framework consisting of four critical aspects: reference agreement, faithfulness, diversity, and utility. For each aspect, we design semantic-based metrics to reflect the evaluation objectives. Meta-evaluation studies demonstrate that our evaluation strategy correlates better with human preferences compared to a range of previously proposed metrics. Using KPEVAL, we re-evaluate 23 keyphrase systems and discover that (1) established model comparison results have blind-spots especially when considering reference-free evaluation; (2) large language models are underestimated by prior evaluation works; and (3) there is no single best model that can excel in all the aspects.

1 Introduction

Building automated keyphrase prediction systems has been a long-lasting research interest of Information Retrieval (IR) and Natural Language Processing (NLP) (Witten et al., 1999; Hulth, 2003; Meng et al., 2017). While a large number of keyphrase prediction systems have been proposed, the majority of them are assessed using a simplistic method: comparing the stemmed predictions with human references for exact matches. An extensive review of 76 recent keyphrase extraction and generation papers published in major conferences reveals a predominant reliance on exact matching, with 75 of 76 papers employing it and 51 treating it as the sole evaluation criterion (Appendix A).

This over-reliance brings two major concerns. First, it has been established that the *evaluation accuracy* of exact matching is inadequate (Zesch and Gurevych, 2009). Although a number of heuristics are proposed to relax the matching criteria (Zesch and Gurevych, 2009; Kim et al., 2010; Luo et al., 2021; Koto et al., 2022) or enrich the label set (Chan et al., 2019), they still struggle to accurately capture *phrase semantics* and have not been validated by systematic meta-evaluation. Second, solely relying on reference-based evaluation is an *incomplete* strategy that overlooks critical aspects such as diversity of the predicted keyphrases (Bahuleyan and El Asri, 2020) or their utility in practical applications of keyphrase systems such as indexing for IR applications (Boudin and Gallina, 2021).

In this paper, we undertake a systematic approach to advance keyphrase evaluation. For reference-based evaluation, we propose a phrase-level semantic matching metric with a high quality embedding trained on large-scale keyphrase data. Based on the human evaluation corpus annotated on KP20k (Meng et al., 2017) and KPTimes (Gallina et al., 2019), the meta-evaluation on five keyphrase systems shows that our metric significantly outperforms existing metrics by more than 0.15 absolute points in Kendall's Tau. By contrast, many proposed improvements to exact matching surprisingly fail to improve its human agreement (§5.3). Further analyses reveal that the proposed metric exhibits enhanced stability under the label variations commonly present in the keyphrase annotations.

Next, we move beyond reference agreement and holistically consider the desiderata for evaluating keyphrase systems. Three crucial aspects are introduced: (1) *faithfulness*, whether the predictions are grounded to the document (§6.1); (2) *diversity*, whether the predictions represent distinct concepts (§6.2); and (3) *utility* for downstream IR applications (§6.3). To accurately assess each aspect, we propose semantically-oriented metric designs
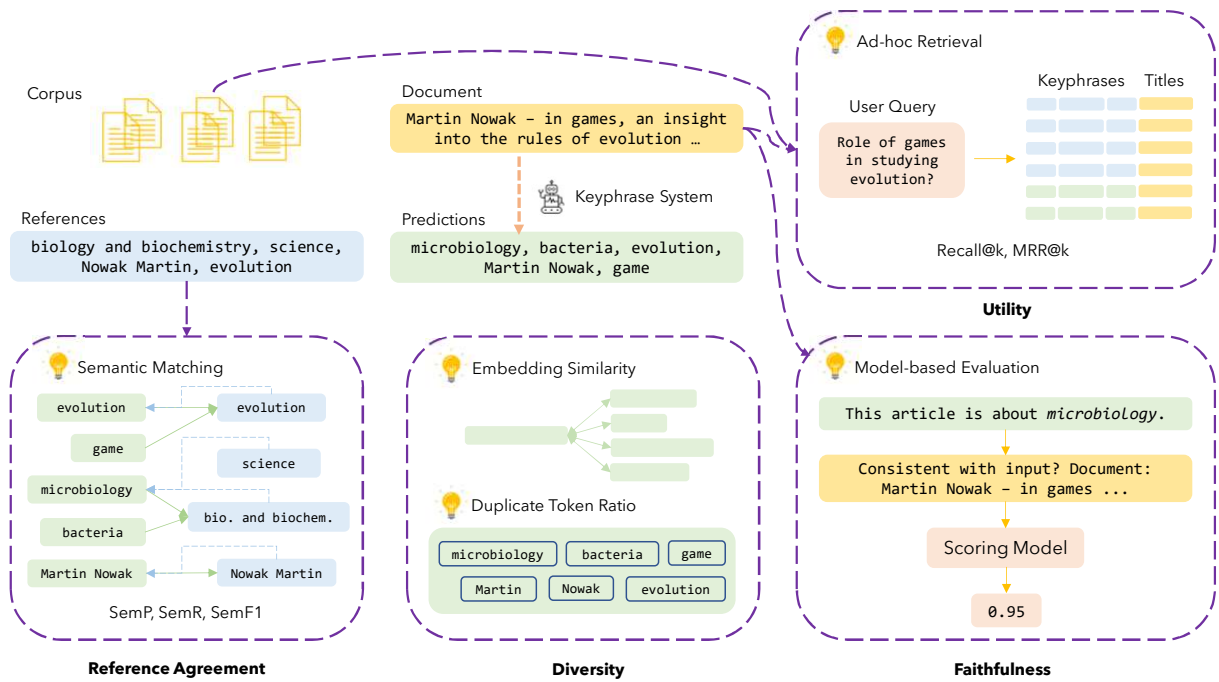
Figure 1: An illustration of the proposed keyphrase evaluation framework. KPEVAL evaluates keyphrase systems on four crucial properties and incorporates semantic-based metrics for more accurate assessment.

including embedding similarity, model-based consistency evaluation, and dense retrieval.

Together, these aspects and metrics form KPE-VAL, a *fine-grained semantic-based* keyphrase evaluation framework (Figure 1). In §7, we employ KPEVAL to evaluate 23 keyphrase systems, producing intriguing insights:

1. KPEVAL uncovers *blind-spots* in established model comparisons, such as the actual superiority of ExHiRD-h (Chen et al., 2020) in many aspects as well as a common difficulty to outperform a baseline in all the aspects.

2. We find that *large language models* (LLMs), particularly GPT-3.5 (Ouyang et al., 2022), exhibit remarkable performance compared to current state-of-the-art keyphrase generation and extraction models. Our results challenge existing conclusions and lead to a reconsideration of using LLMs as keyphrase systems.

3. Finally, KPEVAL's four aspects test distinct abilities at which different models excel, suggesting the importance of aligning evaluation with the diverse needs of real applications.

In summary, KPEVAL establishes a new standard for keyphrase evaluation by advancing reference-based evaluation accuracy and aligning model development with application values via holistic reference-free evaluation. To facilitate future studies, the implementation is released as a toolkit along with the meta-evaluation annotations at https://github.com/uclanlp/KPEval.

2   Related Work

In this section, we review the relevant literature on evaluating keyphrase systems.

Reference-based evaluation   The major metrics for evaluating keyphrase systems are precision, recall, and F1 based on exact-match between the stemmed predictions and references (Mihalcea and Tarau, 2004; Meng et al., 2017; Yuan et al., 2020). This method indiscriminately penalizes unmatched predictions, including synonyms or parent/child concepts of the reference. Later works attempt to improve the metric by *relaxing the matching criterion*. Zesch and Gurevych (2009) propose to use R-precision with approximate matching, tolerating a prediction to be a substring of a reference and vice versa. Kim et al. (2010) employ n-gram matching metrics such as BLEU (Papineni et al., 2002) and Rouge (Lin, 2004). Chan et al. (2019) expand the references with name variations. Luo et al. (2021) propose a fine-grained score that combines token-level matching, edit distance, and duplication penalty. Koto et al. (2022) and Glazkova and Morozov (2022) use the semantic-based BertScore (Zhang et al., 2020) with predictions and references concatenated into two strings.

Meanwhile, ranking-based metrics such as Mean Reciprocal Rank, mean Averaged Precision, and Normalized Discounted Cumulative Gain are introduced to evaluate the ranking provided by keyphrase extraction models (Florescu and

1960

2

Caragea, 2017; Boudin, 2018; Kim et al., 2021). These metrics also compute exact matching to the references during their evaluation.

Reference-free evaluation Directly evaluating keyphrase predictions without references is less common. Early studies conduct human evaluation (Barker and Cornacchia, 2000; Matsuo and Ishizuka, 2004). Later work evaluates the predictions' utility in applications such as retrieval (Bracewell et al., 2005; Boudin and Gallina, 2021) or summarization (Litvak and Last, 2008). Bahuleyan and El Asri (2020) conduct reference-free evaluation of the predictions' diversity.

Meta-evaluation Meta-evaluation studies that compare keyphrase metrics with human evaluations have been limited in scope, with a focus on reference-based evaluation. Kim et al. (2010) compare five lexical matching metrics and concluded that R-precision has the highest Spearman correlation with human judgments. Bougouin et al. (2016) annotated a meta-evaluation corpus with 400 documents in French, evaluating 3 keyphrase models on "appropriateness" and "silence", approximately corresponding to precision and false negative rate.

Discussion Building upon existing literature, this work systematically rethinks the goals of keyphrase evaluation and advances the evaluation methodology. We introduce KPEVAL, a holistic evaluation framework encompassing four key aspects (§4). KPEVAL incorporates semantic-based metrics validated via rigorous meta-evaluation (§5.3 and §6.1). Finally, we conduct a large-scale evaluation of 21 keyphrase systems and offer novel insights into existing model comparisons and LLMs (§7).

3 Background

This section formulates the keyphrase prediction and evaluation tasks and outlines the scope of study.

3.1 Keyphrase Prediction

We denote an instance of keyphrase prediction as a tuple $(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X}$ represents an input document and $\mathcal{Y} = \{y_1, ..., y_n\}$ is a set of n reference keyphrases provided by humans. Each y_i is categorized as a *present keyphrase* if it corresponds to contiguous word sequences in $\mathcal{X}$ after stemming, or an *absent keyphrase* if it does not. Keyphrase generation (KPG) assumes $\mathcal{Y}$ to include both present and absent keyphrases, whereas keyphrase extraction (KPE) only allows present keyphrases in $\mathcal{Y}$.

3.2 Keyphrase Evaluation

The *keyphrase evaluation* process can be viewed as mapping a 4-element tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{C})$ to a real number via a function f. $\mathcal{P} = \{p_1, ..., p_m\}$ is a *set* of m predictions made by a model $\mathcal{M}$ on $\mathcal{X}$. Different from the commonly followed works (Meng et al., 2017; Yuan et al., 2020), we do not distinguish between present and absent keyphrases. This enables matching a predicted keyphrase to any semantically relevant reference, and vice versa.

$\mathcal{C}$ is a corpus that represents the domain of interest, which is an important factor in assessing keyphrase quality. For example, "sports games" may be informative in a general news domain but less so in the specialized domain of basketball news[1]. In this paper, $\mathcal{C}$ will play a crucial role in evaluating the utility of keyphrases for facilitating ad-hoc in-domain document retrieval (§6.3).

3.3 Evaluation Scope

Models This paper covers 21 representative, strong, and diverse keyphrase prediction models spanning three categories: (1) KPE models, (2) KPG models, and (3) large language models (LLMs) and APIs. We aim to include highly cited (up to February 2024) models such as MultipartiteRank (Boudin (2018), 219 citations), CatSeq (Yuan et al. (2020), 92 citations), and SetTrans (Ye et al. (2021), 65 citations). We provide introductions and implementation details in Appendix B.

Datasets We test on two datasets throughout the paper: (1) KP20k (Meng et al., 2017) that features 20k Computer Science papers with keyphrases extracted from the paper metadata and (2) KPTimes (Gallina et al., 2019) that provides 10k news documents paired with keyphrases assigned by expert editors. The two datasets are selected due to their large training sets (500k for KP20k and 250k for KPTimes) and their wide usage in keyphrase research. As such, the models' performance is easier for the community to relate to and the reproduction correctness can be verified more easily.

4 KPEVAL: Evaluation Aspects

We introduce KPEVAL, a fine-grained framework for keyphrase evaluation. KPEVAL posits to evaluate $\mathcal{P}$'s quality across four crucial aspects:

1. **Reference Agreement**: Evaluates the extent to which $\mathcal{P}$ aligns with human-annotated $\mathcal{Y}$.

---

[1]Check Tomokiyo and Hurst (2003) for more examples.

	KP-Set	Input	Reference	Corpus
Reference Agreement	✔		✔	
Faithfulness		✔		
Diversity	✔			
Utility	✔	✔		✔

Table 1: Assumptions of KPEVAL's aspects: whether they operate on a set of keyphrases (KP-Set) and whether they require input, reference, or a corpus.

2. **Faithfulness**: Determines whether each p_i in $\mathcal{P}$ is semantically grounded in $\mathcal{X}$.
3. **Diversity**: Assesses whether $\mathcal{P}$ includes diverse keyphrases with minimal repetitions.
4. **Utility**: Measures the potential of $\mathcal{P}$ to enhance downstream applications, such as document indexing for improved IR performance.

Table 1 outlines the assumptions of the evaluated aspects: whether they are calculated on a set of phrases and whether $\mathcal{X}$, $\mathcal{Y}$, or $\mathcal{C}$ is needed for evaluation. By design, these aspects have deep groundings in the previous literature. Faithfulness and reference agreement can be seen as different definitions of informativeness: the former enforces the information of p_i to be contained in $\mathcal{X}$, while the latter measures $\mathcal{P}$'s coverage of $\mathcal{X}$'s salient information with respect to a background domain (Tomokiyo and Hurst, 2003). Diversity (Bahuleyan and El Asri, 2020) and IR-based utility (Boudin and Gallina, 2021) reflect major efforts to move beyond reference-based evaluation. Building upon these works, KPEVAL aims to provide a unified perspective and to advance the evaluation methodology. Figure 1 illustrates the evaluation design for each aspect, which we will introduce next.

5 KPEVAL: Reference-Based Evaluation with Semantic Matching

To begin with, we focus on reference agreement, the most extensively investigated aspect. Recognizing the limitations of previous approaches, we introduce a semantic matching formulation and conduct meta-evaluation to confirm its effectiveness.

5.1 Reference Agreement: Metric Design

Desiderata: a prediction should be credited if it is semantically similar to a human-written keyphrase; matching should be at phrase-level.

Despite the prevalent use of existing reference-based metrics, their designs harbor intrinsic limitations. On one hand, $F1@5$ (Meng et al., 2017) and $F1@M$ (Yuan et al., 2020) fail to credit many semantically correct predictions. On the other hand,

BertScore with concatenated predictions and references (Koto et al., 2022) reflects semantic similarity, but its token-level matching strategy obscures the semantics of individual keyphrases. Recognizing these limitations, we propose a *phrase-level semantic matching* strategy in KPEVAL and define semantic precision ($SemP$), recall ($SemR$), and F1 ($SemF1$) as follows[2]:

$$SemP(\mathcal{P}, \mathcal{Y}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \max_{y \in \mathcal{Y}} sim(p, y),$$

$$SemR(\mathcal{P}, \mathcal{Y}) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \max_{p \in \mathcal{P}} sim(p, y),$$

$$SemF1(\mathcal{P}, \mathcal{Y}) = \frac{2 \cdot SemP(\mathcal{P}, \mathcal{Y}) \cdot SemR(\mathcal{P}, \mathcal{Y})}{SemP(\mathcal{P}, \mathcal{Y}) + SemR(\mathcal{P}, \mathcal{Y})},$$

where sim is the similarity between the representation of two phrases. To enable the use of any existing dense embedding model, in this paper, we operationalize sim with the cosine similarity:

$$sim(p, q) = cos\_sim(h_p, h_q) = \frac{h_p^T h_q}{||h_p|| \cdot ||h_q||},$$

where h_p is the representation of phrase p obtained by aggregating the representation of all tokens in the phrase. To obtain a high quality embedding that captures phrase-level semantics well, we fine-tune a paraphrase model from Reimers and Gurevych (2019)[3] using unsupervised SimCSE (Gao et al., 2021) on 1.04 million keyphrases from the training sets of KP20k, KPTimes, StackEx (Yuan et al., 2020), and OpenKP (Xiong et al., 2019). The data covers a wide range of domains including science, news, forum, and web documents. At inference time, a single phrase p is provided as the input to the model, and the last hidden states are mean-pooled to obtain h_p. We further document the training details of this model in Appendix C.1.

5.2 Meta-Evaluation Setup

We conduct rigorous meta-evaluation to compare $SemF1$ with existing metrics. We sample 50 documents from the test sets of KP20k and KPTimes each. For each document, we obtain predictions from five representative models: MultipartiteRank, CatSeq, SetTrans, in-domain BART models from Wu et al. (2023a), as well as five-shot prompting GPT-3.5[4]. This variety encompasses both KPE

---

[2]Out metric should be distinguished from Bansal et al. (2022). We keep the name choice as the tasks are different.

[3]We use the checkpoint at https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

[4]We use SciBART+OAGKX for KP20k and KeyBART for KPTimes to represent in-domain BART models. Detail regarding the evaluated models are provided in Appendix B.

	Reference Agreement			Faithfulness		
	$p_i \to \mathcal{Y}$	$y_i \to \mathcal{P}$	average	PKP	AKP	all
KP20k	0.735	0.750	0.743	0.644	0.526	0.627
KPTimes	0.788	0.741	0.765	0.606	0.514	0.563

Table 2: Inter-annotator agreement measured via the interval Krippendorff's alpha. $\to$ denotes the direction of matching a single phrase against a set of phrases. "PKP" and "AKP" denote present and absent keyphrases.

and KPG model, and includes unsupervised, supervised, and few-shot prompting methods. Then, three crowd-source annotators are asked to rate on Likert-scale the *semantic similarity* between (1) each prediction keyphrase p_i and the most semantically similar keyphrase in $\mathcal{Y}$, and (2) each reference keyphrase y_i and the most semantically similar keyphrase in $\mathcal{P}$. We report the details of the annotator recruitment process, the annotation instructions, and the interface in Appendix D.

A total of 1500 document-level annotations with 13401 phrase-level evaluations are collected. As presented in Table 2, we observe 0.75 Krippendorff's alpha for both datasets and both matching directions, indicating a high inter-annotator agreement. The annotations are aggregated to obtain (1) *phrase-level* scores for matching a single phrase to a set of phrases ($p_i \to \mathcal{Y}$ and $y_i \to \mathcal{P}$) and (2) *document-level* precision, recall, and F1 scores, calculated after normalizing the scores to a 0-1 scale.

5.3 Meta-Evaluation Results

Using the *document-level* F1 score annotations, we compare $SemF1$ with six baseline metrics:

1. Exact Matching $F1@M$ (Yuan et al., 2020).
2. $F1@M$ with Substring Matching. We conclude a match between two phrases if either one is a substring of the other. This corresponds to the INCLUDES and PARTOF strategy in Zesch and Gurevych (2009).
3. R-precision (Zesch and Gurevych, 2009).
4. *FG* (Luo et al., 2021).
5. Rouge-L $F1$ (Lin, 2004).
6. BertScore $FScore$ (Zhang et al., 2020)[5]. We concatenate all the phrases in $\mathcal{P}$ with commas to form a single prediction string, and do the same for $\mathcal{Y}$ to form the reference string[6].

We apply Porter Stemmer (Porter, 1980) on $\mathcal{P}$ and $\mathcal{Y}$ before calculating baseline 1, 2, and 3.

---

[5] We use the RoBERTa-large model and the representation at the 17th layer, as recommended by the official implementation at https://github.com/Tiiiger/bert_score.

[6] We find that BertScore is insensitive to the order of labels and predictions. Details are discussed in Appendix C.4.
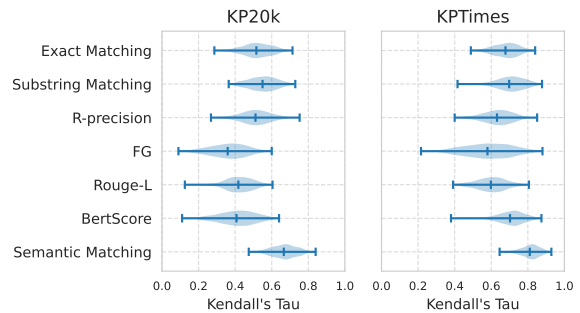

Figure 2: The 95% confidence intervals for the Kendall's Tau between human and automatic metrics on KP20k and KPTimes. $SemF1$ exhibits a higher correlation with humans and smaller intervals.

	KP20k			KPTimes		
	r	ρ	τ	r	ρ	τ
Exact Matching F1@M	0.705	0.689	0.590	0.768	0.792	0.684
FastText	0.808	0.797	0.635	0.844	0.850	0.693
Phrase-BERT	0.792	0.793	0.631	0.876	0.898	0.755
SpanBERT	0.757	0.724	0.562	0.824	0.819	0.653
SimCSE (unsupervised)	0.862	0.834	0.677	0.906	0.905	0.762
SimCSE (supervised)	0.844	0.827	0.669	0.907	0.915	0.780
Ours	**0.898**	**0.884**	**0.735**	**0.926**	**0.925**	**0.794**
- fine-tuning	0.870	0.856	0.705	0.911	0.915	0.780

Table 3: A comparison between different phrase embedding models on scoring an individual phrase against a set of phrases. Our model achieves the best matching correlation with humans, significantly outperforming the second best with $p < 0.001$ via a paired t-test.

In Figure 2, we report the 95% confidence interval of Kendall's Tau via input-level bootstrap resampling with 1000 samples, following Deutsch et al. (2021). Surprisingly, although exact matching produces many false negatives, *existing proposals to relax exact matching do not provide much overall performance gains* either: while substring matching consistently outperforms exact matching by a small amount, R-precision and FG have a lower correlation with human compared to exact matching. BertScore's performance is highly domain-dependent: it achieves the second-best performance on KPTimes while performs poorly on KP20k. By contrast, $SemF1$ greatly outperforms other metrics on both datasets, with a much higher mean score and a smaller variation.

Is the observed high performance of $SemF1$ consistent with any embedding model? Our ablation studies suggest a negative answer. We evaluate with fine-grained *phrase-level annotations* for both directions of matching (i.e., $p_i \to \mathcal{Y}$ and $y_i \to \mathcal{P}$). Table 3 presents the Pearson correlation (r), Spearman correlation (ρ), and Kendall's Tau (τ) of exact matching and semantic matching with various embedding models: FastText (Joulin

et al., 2016)[7], Phrase-BERT (Wang et al., 2021), SpanBERT (Joshi et al., 2020), SimCSE (Gao et al., 2021)[8], our phrase embedding, and the model before the proposed fine-tuning. Despite being a strong strategy by design, semantic matching fails to outperform exact matching in Kendall's Tau with SpanBERT embedding. With the proposed model, semantic matching outperforms exact matching by 0.1 absolute points in Kendall's Tau and more than 0.15 absolute points in Pearson and Spearman correlation. It is worth-noting that although the base SBERT model already achieves strong performance, our phrase-level contrastive fine-tuning provides further performance gains.

Remark We have confirmed that the semantic matching strategy better accommodates semantically correct *predictions*. Additionally, our preliminary study indicates that human references often exhibit lexical variability. When faced with such variability, $SemF1$ demonstrates lower variance than $F1@M$ (detailed in Appendix C.5).

6 KPEVAL: Reference-Free Evaluation

For a range of text generation tasks, the optimal output is often highly aspect-specific (Wen et al., 2015; Mehri and Eskenazi, 2020; Fabbri et al., 2021). As such, reference-based evaluation is incomplete as it does not always align with the evaluation goals. To address this gap, KPEVAL introduces three novel evaluation aspects, along with corresponding reference-free metrics, aimed at aligning closer with real-world application requirements.

6.1 Faithfulness

> *Desiderata: keyphrase predictions should always be grounded in the document.*

In practical scenarios, it is vital for keyphrase systems to refrain from producing concepts not covered in the document, which we term as unfaithful keyphrases. Determining whether a keyphrase is faithful is non-trivial: an absent keyphrase could be faithful by being synonyms or parent/child concepts of the concepts in the document, while a present keyphrase could be deemed unfaithful if it has a wrong boundary. For instance, the keyphrase "hard problem" is unfaithful to a document discussing "NP-hard problem". This example also

---

ID	Model	KP20k	KPTimes
M4	MultipartiteRank	0.694	0.829
M10	CatSeq	0.722	0.936
M15	SetTrans	0.777	0.912
M18	KeyBART	-	0.933
M19	SciBART-large+OAGKX	0.779	-
M21	text-davinci-003 (5-shot)	0.750	0.966

Table 4: Faithfulness evaluation of the predictions made by five models. We report the proportion of keyphrases marked as faithful by human annotators.

illustrates the inadequacy of reference-based evaluation, as "hard problem" may achieve a high score when matched against "NP-hard problem".

Are existing keyphrase models faithful? We conduct a human evaluation of the same set of five models in §5.2 on 100 documents from KP20k and KPTimes each. For each (document, keyphrase prediction) pair, three annotators are asked to make a binary judgement between faithful and unfaithful (details in Appendix D). Table 2 presents the inter-annotator agreement. We find a moderate agreement for present keyphrases and a lower agreement for absent keyphrases. We aggregate the present keyphrase annotations by majority voting. For the absent keyphrases, two of the authors manually resolve the instances where the crowd-source annotators do not agree. Table 4 presents the faithfulness scores for the evaluated models. Surprisingly, M3's outputs are not as faithful as the neural KPG models, supporting the hypothesis that extractive models can suffer from the boundary mistakes that harm their faithfulness. In addition, models make more unfaithful predictions in KP20k compared to KPTimes, indicating the possible difficulty of accurately generating concepts grounded in scientific papers compared to news documents.

Automatic Faithfulness Evaluation Using the human annotations, we evaluate three automatic metrics for judging a keyphrase's faithfulness:

1. The precision metric of **BertScore**$(\mathcal{X}, p_i)$. We use the RoBERTa-large model as in §5.3.
2. The faithfulness metric of **BartScore** (Yuan et al., 2021). p_i is embedded into "*In summary, this is a document about p_i*" for calculating its probability given $\mathcal{X}$. BART-large trained on CNN-DM (See et al., 2017) is used.
3. The consistency metric of **UniEval** (Zhong et al., 2022), which scores text generation as boolean QA. We embed $\mathcal{X}$ and p_i with a

	KP20k	KPTimes
BertScore	0.676	0.648
BartScore	0.677	0.663
UniEval	**0.690**[†]	**0.672**[†]

Table 5: Meta-evaluation results for faithfulness metrics. We report the AUROC evaluated with human annotations in Table 4. [†]statistically significantly better than the second best with $p < 0.01$ via a paired t-test.

template for summarization evaluation: "*question: Is this claim consistent with the document? </s> summary: the document discusses about p_i. </s> document: $\mathcal{X}$*". Then, we use the UniEval model for summarization evaluation provided by the original authors to obtain a score expressed as the probability of the model generating "Yes" normalized by the probability for "Yes" and "No".

All of these metrics output a real number score. To compare their performance, we report their AUROC in Table 5. On both datasets, UniEval outperforms BertScore and BartScore, achieving the highest agreement with human raters. Currently, KPEVAL adopts UniEval as the default faithfulness metric. We encourage future work to continue developing stronger metrics for this aspect.

6.2 Diversity

> *Desiderata: reward more semantically distinct concepts and penalize repetitions.*

Generating keyphrases with minimal repetition is a desirable property of keyphrase applications. To assess the diversity of $\mathcal{P}$, KPEVAL includes one lexical and one semantic metric based on Bahuleyan and El Asri (2020). The lexical metric dup_token_ratio is the percentage of duplicate tokens after stemming. The semantic metric dup_emb_sim is the average of pairwise cosine similarity, using the phrase embedding in §5.1:

$$emb\_sim(\mathcal{P}) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{1}(i \neq j) sim(p_i, p_j)}{m(m-1)}.$$

We note that by design, we do not penalize overgenerating uninformative keyphrases, as it intuitively implies a high diversity[9]. Judging the quality of the keyphrases is instead delegated to the metrics for reference agreement and faithfulness.

6.3 Utility

> *Desiderata: reward predictions that facilitate effective ad-hoc retrieval of the document.*

Information Retrieval (IR) is an important downstream application for keyphrases (Jones and Staveley, 1999; Song et al., 2006; Kim et al., 2013). To directly evaluate whether $\mathcal{M}$ can generate useful keyphrases for IR-related tasks, KPEVAL tests $\mathcal{P}$ on facilitating *ad-hoc retrieval* of $\mathcal{X}$ from an in-domain corpus $\mathcal{C}$ (Boudin and Gallina, 2021).

Concretely, we leverage an in-domain corpus $\mathcal{C}$ that has documents and human-annotated keyphrases[10]. We first index $\mathcal{C}$'s documents into the form *(title, keyphrases) → document*. To evaluate $\mathcal{P}$, we add a single entry *($\mathcal{X}$'s title, $\mathcal{P}$) → $\mathcal{X}$* to the aforementioned database. Then, a set of queries $\mathcal{Q} = \{q_1, q_2, ..., q_{|Q|}\}$ specifically written for $\mathcal{X}$ are used to attempt to retrieve $\mathcal{X}$ from this pool. The utility of $\mathcal{P}$ is measured by two metrics for retrieval effectiveness: Recall at k ($Recall@k$) and Reciprocal Rank at k ($RR@k$), averaged across all queries in $\mathcal{Q}$. To simulate the queries written by real users, we use GPT-4 (OpenAI, 2023) to annotate three adhoc queries based on each document. For KP20k, the queries are in the style of in-text citations similar to Boudin and Gallina (2021). For KPTimes, we generate short phrases that mimic queries on web search engines. We present the prompting details in Appendix C.2. For metric calculation, we consider BM25 (Robertson and Walker, 1994) and a dense embedding model[11] as the retriever and report the averaged scores.

7 Fine-grained benchmarking of keyphrase systems

Finally, we benchmark 23 keyphrase systems with KPEVAL, with the full evaluation results on KP20k and KPTimes presented in Table 7. The implementation details are presented in Appendix B. This section presents the insights on two questions:

1. Do our finding align with the conclusions drawn from previous model comparisons?
2. How do large language models compare with existing keyphrase prediction methods?

Uncovering blind-spots of established results We revisit a set of models compared in Ye et al.

---

[9]As a result, metrics such as the orthogonal regularization term used by CatSeqD (Yuan et al., 2020) are not suitable for our purposes, as the term naturally increases with $|\mathcal{P}|$.

[10]We use the respective training sets as $\mathcal{C}$ for KP20k and KPTimes. In practice, one can run any keyphrase prediction method if human-written keyphrases are not available for $\mathcal{C}$.

[11]We use `cross-encoder/ms-marco-MiniLM-L-6-v2` model via huggingface.

Model	#KP	Reference Agreement (↑)			Faithfulness (↑)	Diversity (↓)		Utility (↑)	
		$F1@M(P)$	$F1@M(A)$	$SemF1$	$UniEval$	dup	emb_sim	$RR@5$	$Recall@5$
(A) CatSeq	7.2	0.362	0.003	0.548	0.774^{BD}	0.370	0.310	0.884^{C}	0.797^{C}
(B) CatSeqTG-2RF1	7.9	0.386^{AC}	0.005^{A}	0.553^{A}	0.734	0.355^{A}	0.251^{A}	0.886^{C}	0.801^{AC}
(C) ExHiRD-h	5.5	0.374^{A}	0.005^{A}	0.562^{AB}	0.781^{ABD}	0.214^{ABD}	0.195^{AB}	0.878	0.791
(D) SetTrans	7.7	0.390^{AC}	0.006^{ABC}	0.583^{ABC}	0.766^{B}	0.308^{AB}	0.203^{AB}	0.889^{AC}	0.803^{AC}

Table 6: Re-evaluation of four models considered in Ye et al. (2021). #KP = Number of keyphrases. (P) and (A) indicate present and absent keyphrases. $dup = dup\_token\_ratio$. For dup and emb_sim, lower scores are better. We use four superscripts to mark the results that are significantly better than CatSeq (**A**), CatSeqTG-2RF1 (**B**), ExHiRD-h (**C**), and SetTrans (**D**) with $p < 0.01$ via a paired t-test.
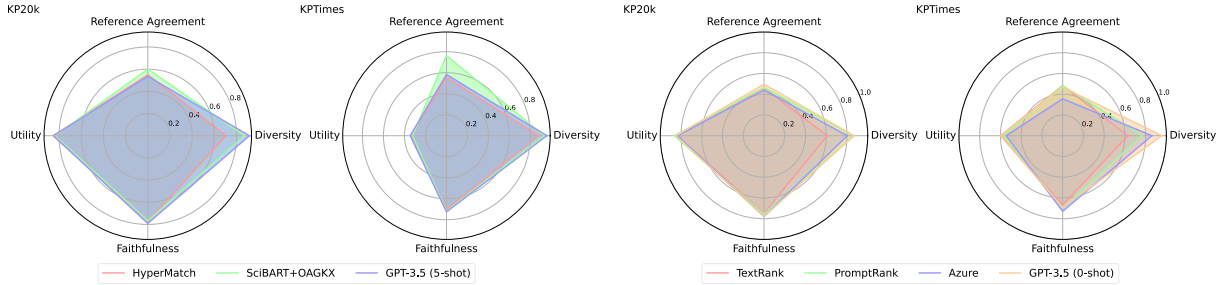

Figure 3: A comparison between GPT-3.5 and strong supervised (left) and unsupervised (right) keyphrase extraction and keyphrase generation methods. GPT-3.5 achieves strong performance on both datasets and most dimensions. We use $RR@5$ to represent utility and $1 - dup\_token\_ratio$ to represent diversity.

(2021): CatSeq, CatSeqTG-2RF1, ExHiRD-h, and SetTrans. As shown in Table 6, a nuanced pattern emerges when evaluating beyond $F1@M$, the main metric reported in the original paper. $SemF1$ is consistent with $F1@M$ in recognizing SetTrans as the best model for reference agreement. However, SetTrans does not outperform all the three baselines in reference-free evaluation. Specifically, the best faithfulness and diversity scores are achieved by ExHiRD-h, and the difference between SetTrans and CatSeqTG-2RF1 in utility is insignificant. Moreover, contradicting with $F1@M$, KPEVAL's metrics show a superiority of ExHiRD-h over CatSeqTG-2RF1 for reference agreement, faithfulness, and diversity. We provide several supporting examples in Appendix E. By revealing these blind-spots in previous results, KPEVAL enables a holistic view in model comparison and a stricter criterion in establishing the state-of-the-art.

LLM vs traditional keyphrase models With the ascent of LLMs as foundational elements in NLP, their efficacy in keyphrase prediction warrants examination. Prior research reported significant performance gaps when evaluating LLMs with exact matching $F1@M$ (Song et al., 2023; Martínez-Cruz et al., 2023). With KPEVAL, we conduct a more comprehensive investigation. In Figure 3, we compare GPT-3.5 with state-of-the-art KPE and KPG methods. For supervised methods, perfor-

mance of five-shot prompting GPT-3.5 is comparable or better than HyperMatch along every dimension, and comparable to SciBART+OAGKX in diversity, utility, and faithfulness. In addition, zero-shot prompting outperforms the unsupervised TextRank, PromptRank, and the Azure API in diversity while being competitive across other dimensions. These results suggest that the potential of LLMs for keyphrase prediction may be underappreciated under traditional evaluation paradigms.

Discussion What have we learned in this re-evaluation? First, the refined evaluation facilitated by KPEVAL challenges some of the existing model comparisons and emphasizes the difficulty of outperforming baselines across all aspects. In fact, these aspects test unique abilities, exhibiting weak cross-aspect correlations (Appendix C.3) and distinct preferences for keyphrase systems (Table 6, 7). Our findings advocate for a tailored approach to metric weighting, allowing users to customize evaluations based on their evaluation desiderata. Finally, our results reveal strong performance of GPT-3.5, encouraging future work to further understand and improve LLMs for keyphrase prediction.

8 Conclusion

We introduce KPEVAL, a fine-grained evaluation framework that conducts semantic-based evaluation on reference agreement, faithfulness, diversity,

ID	Model	#KP	Reference Agreement			Faithfulness	Diversity		Utility	
			SemP ↑	SemR ↑	SemF1 ↑	UniEval ↑	dup ↓	emb_sim ↓	RR@5 ↑	Recall@5 ↑
KP20k										
M1	TF-IDF*	10.0	0.431	0.524	0.463	0.696	0.396	0.161	0.889	0.805
M2	TextRank*	10.0	0.432	0.518	0.460	0.737	0.391	0.180	0.882	0.796
M3	RAKE*	10.0	0.401	0.499	0.437	0.733	0.295	0.170	0.855	0.783
M4	MultipartiteRank*	10.0	0.364	0.548	0.429	0.712	0.148	0.092	0.881	0.794
M5	YAKE!*	10.0	0.434	0.508	0.459	0.778	0.372	0.190	0.859	0.786
M6	PromptRank*✿	10.0	0.414	0.556	0.465	0.755	0.133	0.159	0.866	0.786
M7	Kea*♦	10.0	0.452	0.533	0.479	0.715	0.445	0.175	**0.892**	**0.809**†
M8	BERT+CRF*♦✿	10.0	**0.675**†	0.447	0.515	**0.801**†	0.137	0.399	0.879	0.796
M9	HyperMatch*♦✿	10.0	0.505	**0.631**†	0.549	0.772	0.297	0.233	0.881	0.798
M10	CatSeq♦	7.2	0.596	0.531	0.548	0.774	0.370	0.310	0.884	0.797
M11	CatSeqTG-2RF1♦	7.9	0.550	0.579	0.553	0.734	0.355	0.251	0.886	0.801
M12	ExHiRD-h♦	5.5	0.587	0.559	0.562	0.781	0.214	0.195	0.878	0.791
M13	SEG-Net♦	11.0	0.560	0.596	0.565	0.757	0.260	0.177	0.889	0.804
M14	Transformer♦	8.2	0.558	0.593	0.562	0.747	0.289	0.239	0.878	0.791
M15	SetTrans♦	7.7	0.570	0.623	0.583	0.766	0.308	0.203	0.889	0.803
M16	SciBERT-G♦✿	6.1	0.587	0.590	0.577	0.762	0.152	0.177	0.878	0.791
M17	BART-large♦✿	6.6	0.567	**0.607**	0.574	0.759	0.144	0.168	0.877	0.790
M18	KeyBART♦✿	6.0	0.580	0.599	0.577	0.765	0.130	0.163	0.883	0.792
M19	SciBART-large+OAGKX♦✿	6.1	0.601	0.623	**0.601**†	0.766	0.129	0.158	0.879	0.793
M20	text-davinci-003 (0-shot)✿	9.5	0.433	0.616	0.498	0.777	0.143	0.110	0.888	0.799
M21	text-davinci-003 (5-shot)✿	6.5	0.503	0.599	0.535	0.790	**0.084**†	0.113	0.886	0.802
M22	Amazon Comprehend*	10.0	0.224	0.324	0.259	0.693	0.436	0.103	0.855	0.758
M23	Azure Cognitive Services*	10.0	0.387	0.536	0.440	0.778	0.198	**0.068**†	0.874	0.780
KPTimes										
M1	TF-IDF*	10.0	0.403	0.557	0.461	0.658	0.175	0.163	0.589	0.482
M2	TextRank*	10.0	0.457	0.532	0.483	0.667	0.371	0.246	0.598	0.485
M3	RAKE	10.0	0.284	0.380	0.321	0.693	0.157	0.140	0.555	0.442
M4	MultipartiteRank*	10.0	0.410	0.575	0.472	0.670	0.123	0.150	0.589	0.483
M5	YAKE!	10.0	0.387	0.442	0.405	0.639	0.317	0.225	0.564	0.441
M6	PromptRank*✿	10.0	0.427	0.574	0.483	0.699	0.233	0.134	0.569	0.458
M7	Kea*♦	10.0	0.427	0.575	0.484	0.699	0.219	0.175	**0.607**†	**0.500**†
M8	BERT+CRF*✿	2.3	0.741	0.459	0.550	**0.769**†	0.072	0.556	0.583	0.478
M9	HyperMatch*♦✿	10.0	0.493	0.655	0.554	0.703	0.124	0.271	0.587	0.477
M10	CatSeq♦	5.9	0.723	0.714	0.708	0.683	0.191	0.257	0.569	0.454
M11	CatSeqTG-2RF1♦	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
M12	ExHiRD-h♦	5.8	0.725	0.713	0.709	0.680	0.149	0.238	0.572	0.454
M13	SEG-Net♦	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
M14	Transformer♦	5.7	0.720	0.736	0.717	0.658	0.133	0.231	0.569	0.456
M15	SetTrans♦	8.4	0.662	0.801	0.716	0.645	0.210	0.232	0.572	0.454
M16	SciBERT-G♦✿	4.5	**0.784**	0.737	0.749	0.716	0.064	0.208	0.579	0.468
M17	BART-large♦✿	5.4	0.768	0.796	**0.770**†	0.706	0.074	0.203	0.582	0.469
M18	KeyBART♦✿	5.9	0.751	**0.807**†	0.766	0.696	0.086	0.206	0.579	0.466
M19	SciBART-large+OAGKX♦✿	4.8	0.782	0.766	0.763	0.718	0.069	0.203	0.581	0.472
M20	text-davinci-003 (0-shot)✿	14.1	0.383	0.620	0.467	0.690	0.082	0.137	0.599	0.487
M21	text-davinci-003 (5-shot)✿	6.7	0.549	0.643	0.582	0.725	**0.044**†	0.188	0.590	0.479
M22	Amazon Comprehend*	10.0	0.258	0.347	0.292	0.617	0.291	0.203	0.544	0.433
M23	Azure Cognitive Services*	10.0	0.312	0.425	0.355	0.728	0.140	**0.133**†	0.543	0.425

Table 7: Evaluation results for 23 evaluated keyphrase systems. Due to budget constraints, we only sample 1000 documents per dataset for utility evaluation. For the other aspects, the complete test sets are used. #KP = Number of keyphrases. $dup = dup\_token\_ratio$. We use ↑ for the higher the better and ↓ for the reverse. The best is boldfaced and the second best is underlined. † indicates statistically significantly better than the second best with $p < 0.01$ via a paired t-test. * = KPE systems. ♦ = supervised models. ✿ = pre-trained language models.

and utility of keyphrase systems. We show the advantage of our metrics via rigorous human evaluation, and exhibit the usability of KPEVAL through a large-scale evaluation of keyphrase systems including LLM-based methods and keyphrase APIs. Our framework marks the first step towards systematically evaluating keyphrase systems in the era of LLMs. We hope KPEVAL can motivate future works to adopt more accurate evaluation metrics and further advance the evaluation methodology. Future studies might also explore the development of utility metrics tailored to the specific requirements of applications in niche domains.

Limitations

While our study sheds light on enhancing the keyphrase evaluation methodology, several limitations exist for KPEVAL.

1. **Multilingual Evaluation.** We encourage future work to extend the evaluations in this paper to multilingual setting. By design, the aspect and metric formulations in KPEVAL are language-agnostic. For instance, $SemF1$ can be implemented with multilingual embeddings. Such embeddings need not to keyphrase-specific. For instance, Table 3 suggests that off-the-shelf embeddings cam already have reasonable performance.

2. **Alternative Scoring Schemes.** KPEVAL's evaluation and meta-evaluation strategies always target at producing fine-grained numeric scores. This is different from tasks like machine translation where direct assessment scores are annotated (Graham et al., 2013) or LLM competitions that report Elo scores (Wu et al., 2023b). Exploring whether these schemes may provide better evaluation quality is an important question for future work on keyphrase evaluation.

3. **LLM-Based Evaluation.** Recent works have shown the viability of using LLMs for human-aligning aspect-specific evaluation (Liu et al., 2023). By comprehensively establishing the possible evaluation aspects and curating meta-evaluation data for reference agreement and faithfulness, our work sets up the necessary preparations for evaluating LLM-based metrics. We encourage future work to formally define and investigate the performance of keyphrase evaluation metrics based on LLMs.

Acknowledgments

References

Wasi Ahmad, Xiao Bai, Soomin Lee, and Kai-Wei Chang. 2021. Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention. 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 1389–1404, Online. Association for Computational Linguistics.

Hareesh Bahuleyan and Layla El Asri. 2020. Diverse keyphrase generation with neural unlikelihood training. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5271–5287, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Naman Bansal, Mousumi Akter, and Shubhra Kanti Karmaker Santu. 2022. SEM-f1: an automatic way for semantic evaluation of multi-narrative overlap summaries at scale. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 780–792, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence: 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2000 Montéal, Quebec, Canada, May 14–17, 2000 Proceedings 13*, pages 40–52. Springer.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 667–672, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Adrien Bougouin, Sabine Barreaux, Laurent Romary, Florian Boudin, and Béatrice Daille. 2016. Termitheval: a french standard-based resource for keyphrase extraction evaluation. In *LREC-Language Resources and Evaluation Conference*.

D. Bracewell, F. Ren, and S. Kuriowa. 2005. Multilingual single document keyword extraction for information retrieval. *2005 International Conference on Natural Language Processing and Knowledge Engineering*, pages 517–522.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Inf. Sci.*, 509:257–289.

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

Wang Chen, Hou Pong Chan, Piji Li, Lidong Bing, and Irwin King. 2019. An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. 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 2846–2856, Minneapolis, Minnesota. 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.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anna Glazkova and Dmitry Morozov. 2022. Applying transformer-based text summarization for keyphrase generation. *arXiv preprint arXiv:2209.03791*.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. 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, Berlin, Germany. 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.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

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*, SIGIR '99, page 160–167, New York, NY, USA. Association for Computing Machinery.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Jihyuk Kim, Myeongho Jeong, Seungtaek Choi, and Seung-won Hwang. 2021. Structure-augmented keyphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2657–2667, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. 2010. Evaluating n-gram based evaluation metrics for automatic keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 572–580, Beijing, China. Coling 2010 Organizing Committee.

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.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xiaoyan Bai. 2023. PromptRank: Unsupervised keyphrase extraction using prompt. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9788–9801, Toronto, Canada. Association for Computational Linguistics.

Fajri Koto, Timothy Baldwin, and Jey Han Lau. 2022. LipKey: A large-scale news dataset for absent keyphrases generation and abstractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3427–3437, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, Manchester, UK. Coling 2008 Organizing Committee.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yichao Luo, Yige Xu, Jiacheng Ye, Xipeng Qiu, and Qi Zhang. 2021. Keyphrase generation with fine-grained evaluation-guided reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 497–507, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roberto Martínez-Cruz, Alvaro J López-López, and José Portela. 2023. Chatgpt vs state-of-the-art models: A benchmarking study in keyphrase generation task. *arXiv preprint arXiv:2304.14177*.

Yutaka Matsuo and Mitsuru Ishizuka. 2004. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric

for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

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

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

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

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 40:211–218.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

S. E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, pages 232–241, London. Springer London.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

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, Vancouver, Canada. Association for Computational Linguistics.

Min Song, Il Yeol Song, Robert B. Allen, and Zoran Obradovic. 2006. Keyphrase extraction-based query expansion in digital libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, page 202–209, New York, NY, USA. Association for Computing Machinery.

Mingyang Song, Yi Feng, and Liping Jing. 2022. Hyperbolic relevance matching for neural keyphrase extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5710–5720, Seattle, United States. Association for Computational Linguistics.

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*.

Mingyang Song, Liping Jing, and Lin Xiao. 2021. Importance Estimation from Multiple Perspectives for Keyphrase Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2726–2736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40, Sapporo, Japan. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10837–10851, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

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 Uddin Ahmad, and Kai-Wei Chang. 2022. Pre-trained language models for keyphrase generation: A thorough empirical study.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023a. Rethinking masked language modeling for Chinese spelling correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10743–10756, Toronto, Canada. Association for Computational Linguistics.

Yuxiang Wu, Zhengyao Jiang, Akbir Khan, Yao Fu, Laura Ruis, Grefenstette Edward, and Tim Rocktäschel. 2023b. Chatarena: Multi-agent language game environments for large language models. https://github.com/chatarena/chatarena.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5175–5184, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. 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.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

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, Online. Association for Computational Linguistics.

Torsten Zesch and Iryna Gurevych. 2009. Approximate matching for evaluating keyphrase extraction. In *Proceedings of the International Conference RANLP-2009*, pages 484–489, Borovets, Bulgaria. Association for Computational Linguistics.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on Twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, Texas. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Supplementary Material: Appendices

A Literature survey: evaluation methods used in recent keyphrase papers

We survey all the papers published from 2017 to 2023 in major conferences for AI, NLP, and IR (ACL, NAACL, EMNLP, AAAI, and SIGIR) about keyphrase extraction or keyphrase generation. We choose year 2017 as it marks the start of deep keyphrase generation methods (Meng et al., 2017). We manually check each paper's experiment sections and note down which of the six major categories do the reported evaluation metrics belong to: (1) precision, recall, and F1 based on exact-matching; (2) diversity metrics, such as duplication ratio; (3) ranking-based metrics such as mAP, α-NDCG, and MRR; (4) approximate versions of exact matching such as n-gram matching; (5) retrieval-based utility metrics; (6) human evaluation. We make sure each of metrics used in the surveyed papers can fall under one and only one category under this ontology.

The survey results are presented in Figure 4. Overall, despite its limitation, exact matching has been *de facto* the method for assessing the performance of newly proposed keyphrase systems, and there has been limited progress in adopting alternative metrics. The majority of papers report exact matching precision, recall, and F1. Two thirds of all papers use exact matching as the only metric, including 10 out of 11 papers published in 2023. Human evaluation is only conducted in one paper surveyed (Bennani-Smires et al., 2018).

B Keyphrase Systems: Implementation Details and Full Evaluation Results

In this section, we describe in detail the considered keyphrase systems as well as how we obtain their outputs for evaluation.

B.1 Keyphrase Systems

We consider three types of keyphrase systems: keyphrase extraction models, keyphrase generation models, and APIs including large language models.

Keyphrase Extraction Systems KPE has traditionally been approached through unsupervised methods, where noun phrase candidates are ranked using heuristics (Hulth, 2003; Mihalcea and Tarau, 2004). Supervised approaches include feature-based ranking (Witten et al., 1999), sequence label-
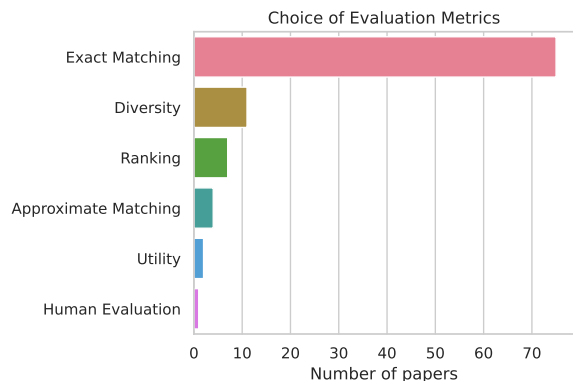

Figure 4: Distribution over 75 papers of the adopted evaluation metrics: (1) F1 score based on exact-matching; (2) diversity metrics such as duplication ratio; (3) ranking-based metrics such as mAP, α-NDCG, and MRR; (4) approximate versions of exact matching; (5) retrieval-based utility metrics; (6) human evaluation.

ing (Zhang et al., 2016), and the use of pre-trained language models (PLMs) for task-specific objectives (Song et al., 2021, 2022). We consider the following nine KPE models:

M1 TF-IDF (Jones, 1972) selects the phrases containing words with highest TF-IDF weight.

M2 TextRank (Mihalcea and Tarau, 2004) runs PageRank (Brin and Page, 1998) on an undirected word cooccurrence graph.

M3 RAKE (Rose et al., 2010) is an efficient single-document unsupervised KPE algorithm that uses the cooccurrence graph to score keyphrase candidates.

M4 MultipartiteRank (Boudin, 2018) represents the document as a multipartite graph to encode topical diversity and improve intra-topic keyphrase selection preferences.

M5 YAKE! (Campos et al., 2020) is an unsupervised KPE method relying on local features such as term co-occurrence and frequencies.

M6 Kea (Witten et al., 1999) builds a supervised keyphrase classifier using statistical features including TF-IDF and position information.

M7 BERT+CRF (Wu et al., 2022) fine-tunes a pre-trained BERT (Devlin et al., 2019) on sequence labeling with conditional random fields (Lafferty et al., 2001).

M8 HyperMatch (Song et al., 2022) trains a supervised model to rank phrase-document relevance in a hyperbolic space.

M9 PromptRank (Kong et al., 2023) ranks the

phrases by their probability given a prompt prefix using a sequence-to-sequence pre-trained language models.

Keyphrase Generation Systems   KPG models are often trained with various supervised objectives, including *One2One*, *One2Seq*, and *One2Set* (Meng et al., 2017; Yuan et al., 2020; Ye et al., 2021). A range of strategies have been proposed, including hierarchical modeling of phrases and words (Chen et al., 2020), reinforcement learning (Chan et al., 2019), unifying KPE and KPG (Chen et al., 2019; Ahmad et al., 2021), and using PLMs (Kulkarni et al., 2022). We consider ten KPG models:

M10 CatSeq (Yuan et al., 2020) is an RNN trained with copy mechanism (Gu et al., 2016) on generating keyphrases as a sequence.

M11 CatSeqTG-2RF1 (Chan et al., 2019) introduces an RL-based approach using recall and F1 as the rewards.

M12 ExHiRD-h (Chen et al., 2020) extends Cat-Seq with a hierarchical decoding and an exclusion mechanism to avoid duplications.

M13 SEG-Net (Ahmad et al., 2021) unifies keyphrase extraction and keyphrase generation training and introduces layer-wise coverage attention mechanism

M14 Transformer (Ye et al., 2021) is a Transformer model (Vaswani et al., 2017) trained with copy mechanism on generating keyphrases as a sequence.

M15 SetTrans (Ye et al., 2021) generates keyphrases in parallel based on control codes trained via a k-step target assignment process.

M16 SciBERT-G (Wu et al., 2022) fine-tunes SciBERT (Beltagy et al., 2019) for seq2seq keyphrase generation using a prefix-LM objective (Dong et al., 2019).

M17 BART-large (Wu et al., 2022) is a BART model (Lewis et al., 2020) fine-tuned on generating keyphrases as a sequence.

M18 KeyBART (Kulkarni et al., 2022) is a BART model adapted to scientific keyphrase generation before fine-tuning.

M19 SciBART-large+OAGKX (Wu et al., 2022) is pre-trained on scientific corpus and scientific keyphrase generation before fine-tuning.

Large language models and APIs.   Recent advancements highlight the capability of large language models (LLMs) to perform in-context learning (Brown et al., 2020). We explore GPT-3.5 (Ouyang et al., 2022) for KPG in the zero-shot

and few-shot prompting setting[12]. We also assess two commercial keyphrase extraction APIs.

M20 Zero-shot prompting GPT-3.5.
M21 Five-shot prompting GPT-3.5.
M22 Amazon Comprehend API
M23 Azure Cognitive Services API

B.2   Implementation Details

We obtain the output from the original authors for M8, M10, M11 for KP20k, and M7, M16, M17, M18 for both KP20k and KPTimes. For the other KPE and KPG models, we reproduce the results on our own. For M1, M2, M4, M6, we obtain the outputs using the pke library. For M3, M5, M8, M10, M11, M12 (KPTimes only) and M9, M13, M14, M15, we use the original implementations provided by the authors. For M18, we use the DeepKPG toolkit. For the commercial APIs, we implement the API call following the instructions. We obtained results on 3/5/2023 for M20 and 3/11/2023 for M21. Following existing KPE literature, we consider the top 10 predictions from M1, 2, 3, 4, 5, 6, 8, 9, 22, and 23. For all the systems, we truncate the input to 512 tokens. We perform hyperparameter tuning on the validation sets and ensure that the models match the performance reported by original paper or existing works such as Wu et al. (2022). For M11 and M13 on KP-Times, we failed to obtain reasonable performance and thus choose to omit the results.

For GPT-3.5, we always start the prompt with a task definition: *"Keyphrases are the phrases that summarize the most important and salient information in a document. Given a document's title and body, generate the keyphrases."*

In the zero-shot setting, we provide the title and body in two separate lines, and start a new line with *"Keyphrases (separated by comma):"*. In the 5-shot setting, we randomly sample 5 examples from the train set for each test document, and provide their title, body, and keyphrases in the same format in the prompt before the document tested.

C   KPEVAL Metrics: Implementation Details and Further Analyses

C.1   Phrase Embedding Training Details

We fine-tune the paraphrase model provided by Reimers and Gurevych (2019) distributed at `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`.

---

[12]We use OpenAI's `text-davinci-003` via API.

Unsupervised SimCSE ([Gao et al., 2021]) is used as the training loss. Specifically, given a batch of B phrases, the loss can be expressed as:

$$\mathcal{L}_{simcse} = \frac{1}{B}\sum_{i=1}^{B} -\log \frac{e^{sim(h_i, h_i')/\tau}}{\sum_{j=1}^{B} e^{sim(h_i, h_j')/\tau}},$$

where h_i and h_i' are the representations of phrase i obtained using two separate forward passes with dropout enabled. This objective discourages the clustering of unrelated phrases in the representation space and retains a high similarity between semantically related phrase pairs. τ is a scaling factor which we empirically set to 0.05.

We fine-tune the model on $\mathcal{L}_{simcse}$ using 1.04 million keyphrases from the training set of KP20k, KPTimes, StackEx ([Yuan et al., 2020]), and OpenKP ([Xiong et al., 2019]), covering a wide range of domains including science, news, forum, and web documents. We use the AdamW optimizer with maximum sequence length 12, batch size 512, dropout 0.1, and learning rate 1e-6 to fine-tune for 1 epoch. The hyperparameters are determined using a grid search on the following search space: batch size {128, 512, 1024, 2048}, learning rate {1e-6, 5e-6, 1e-5, 5e-5}. We randomly hold out 0.5% from the training data for validation and model selection. The final training takes 30 minutes on a single Nvidia GeForce RTX 2080 Ti GPU.

Remark on embedding quality In Table 8, we provide an additional study on the trained embedding. Specifically, following [Gao et al. (2021)]; [Wang and Isola (2020)], we evaluate alignment, the average similarity between keyphrases of similar meanings, and uniformity, the average similarity between unrelated keyphrase pairs. For alignment, we utilize the name-variation pairs constructed by [Chan et al. (2019)]. We find that our model achieves the best uniformity, which means that it assigns close to 0 similarity for unrelated pairs. For phrases with similar meanings, it achieves 0.58 alignment, which is also close to human perceptions. Finally, the separation between uniformity and alignment is also the largest for our embedding model.

C.2 Ad-hoc Query Construction for Utility

We use GPT-4 ([OpenAI, 2023]) to annotate three ad-hoc queries per document from KP20k and KP-Times test sets. For both datasets, we sample with temperature set to 0.9 to balance quality and diversity. Due to budget constraints, we sample 1000 documents per dataset to construct the evaluation set. The prompts are presented in Figure 9.

	Alignment	Uniformity	Δ
Phrase-BERT	**0.78**	0.54	0.24
SpanBERT	0.71	0.54	0.17
Unsup. SimCSE	0.62	0.23	0.39
Sup. SimCSE	0.72	0.41	0.31
SBERT	0.63	0.11	0.52
Ours	0.58	**0.02**	**0.56**

Table 8: A comparison between different phrase embedding models. Our model achieves a large difference between alignment and uniformity, indicating the best ability to distinguish unrelated phrases.

Dataset	Order	r	ρ	τ
KP20k	Default	0.562	0.597	0.436
	Permuted	0.545_4	0.578_5	0.425_4
KPTimes	Default	0.694	0.729	0.553
	Permuted	0.691_3	0.727_3	0.552_3

Table 9: The stability of BertScore when given permuted labels and references. 0.545_4 denotes mean 0.545 with standard deviation 0.004.

C.3 Inter-metric Correlations

Using the document-level evaluation scores of the 21 keyphrase systems, we calculate the pair-wise Kendall's Tau for all the metrics in KPEVAL. The results are shown in Figure 5. Overall, we find that only the metrics for the same dimension show a moderate or strong correlation with each other, and the metrics for different aspects hardly correlate. This results suggest that KPEVAL's aspects measure distinct abilities and that optimizing a single metric does not automatically transfer to a superior performance on the other aspects.

C.4 Order Sensitivity of BertScore

As BertScore is evaluated on two sequences instead of two sets of phrases, previous works concatenate all the predicted phrases as a single string and evaluate against all the references concatenated together ([Koto et al., 2022]; [Glazkova and Morozov, 2022]). However, it is unclear how to order the prediction and reference keyphrases within these two strings and whether BertScore's performance is sensitive to this ordering or not. We conduct phrase-level meta-evaluation of BertScore with phrase-level permutation applied to the matching target. Specifically, we shuffle the labels and the references before calculating meta-evaluation metrics. We repeat this process for 100 times and report the mean and standard deviation of human correlation in Table 9. Overall, we find that the metric-human correlation of BertScore is relatively insensitive to permuta-
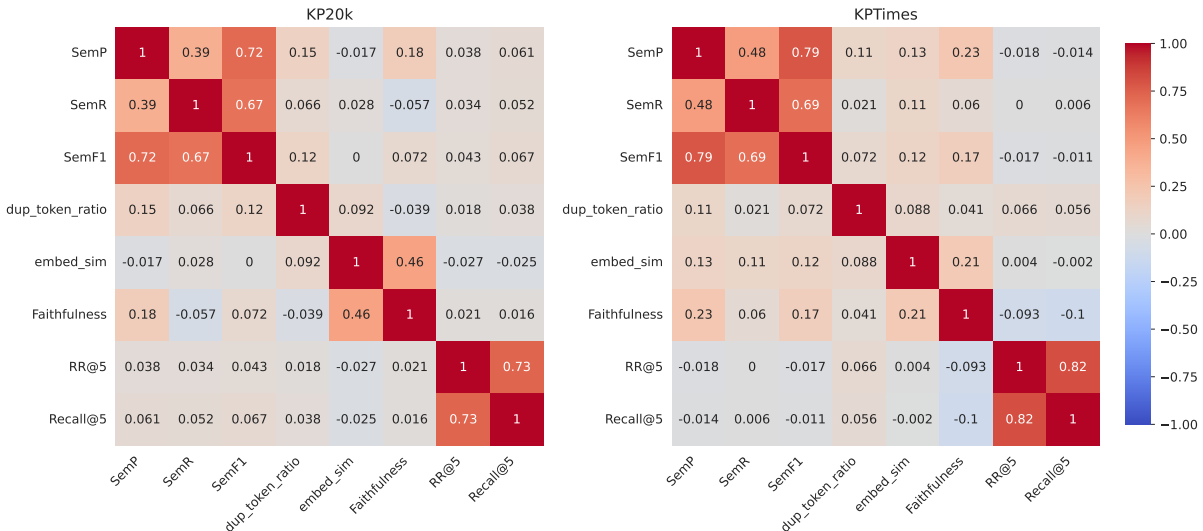
Figure 5: Correlation between KPEVAL's metrics, measured by Kendall's Tau. The diversity scores are negated to provide a more intuitive view. We find that only the metrics for the same dimension correlate with each other.

tions: when given reference phrases or prediction phrases concatenated in different orders, BertScore maintains a similar evaluation quality. One notable pattern is that when the references and the predictions are not permuted, BertScore obtains slightly higher performance. We hypothesize that in this case many phrases may present in the same order in the reference and the prediction, making the exactly matched instances easier to distinguish.

C.5 Variation in Keyphrase Annotations Motivates Semantic Matching

A major motivation for semantic matching is that valid predictions vary in many ways. But at the same time, *do human references also exhibit lexical variations*? We investigate with a pilot study of model-in-the-loop keyphrase annotation.

Setup We sample 100 documents each from the test sets of KP20k and KPTimes and combine each document's $\mathcal{Y}$ with $\mathcal{P}$ from four systems: M8, M10, M15, and M18 (KPTimes only)/M19 (KP20k only). Three MTurk annotators are presented with the document and the phrases re-ordered alphabetically. They are then asked to write keyphrases that best capture the salient information. We state that they may select from the provided keyphrases or write new keyphrases. Figure 12 presents the annotation interface. We use the same set of annotators in appendix D and collect 3226 keyphrase annotations, which approximately cost $700.

Keyphrase annotations exhibit lexical variations. Figure 6 presents the distribution of phrase se-
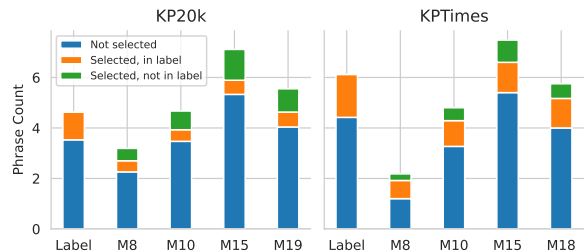

Figure 6: Annotators' selection distribution for each of the keyphrase sources. The reported counts are averaged across three annotators. Annotators do not prefer selecting keyphrases belonging to the original labels.

lected by the annotators from each source. The reported counts are averaged over three annotators. Surprisingly, we find that the keyphrases in the original labels are not preferred over the outputs from other models. First, nearly 70% keyphrases in the original labels are not selected. Second, the annotators select several keyphrases from each model's outputs that do not appear in the label set. For KP20k, the annotators even select more such keyphrases compared to the phrases from the labels. This suggests that *label variations* can be common in keyphrase annotations, even if candidate keyphrases are given as a guidance.

Are the observed label variations caused by annotators writing entirely different concepts? We find that the average character-level edit distance between the selected phrases and the closest phrase in label keyphrases is 11.0 for KP20k and 7.0 for KPTimes, much smaller than the metric for phrases that are not selected (17.5 for KP20k and 14.6 for KPTimes). In other words, keyphrases written by different humans are lexically similar to the origi-

Label Sets	KP20k		KPTimes	
	$F1@M$	$SemF1$	$F1@M$	$SemF1$
Original	0.160	0.498	0.085	0.461
Newly Created 1	0.225	0.536	0.143	0.487
Newly Created 2	0.207	0.500	0.143	0.475
Newly Created 3	0.254	0.556	0.128	0.455
σ	0.039	**0.027**	0.028	**0.014**

Table 10: Evaluating **M20** with $F1@M$ and $SemF1$ using four label sets. $SemF1$ displays a higher consistency across different label versions, as indicated by a lower standard deviation (σ).

nal references, with slightly different phrasings.

Semantic matching is more robust to label variations. A desired property of keyphrase metrics is outputting consistent scores with similar sets of labels. Using the 200 annotated documents, we compare $F1@M$ and $SemF1$ on the outputs from **M20** with four sets of labels[13]: the original labels and three sets of newly annotated labels. As shown in Table 10, the output of $SemF1$ is more stable using different label sets, indicating a higher robustness and reliability compared to $F1@M$.

D Annotation for Meta-Evaluation

Annotator Hiring For all the annotation experiments, we use Amazon Mechanical Turk (MTurk) and designed a qualification task to hire and train annotators. We require the annotators to be located in United States or United Kingdom and have finished more than 1000 HITs with 97% pass rate. In the qualification task, the annotators are presented with the definition of semantic matching with examples, and then asked to annotate three documents. 46 annotators that have average ≤ 1.5 wrong annotations per document are selected. We ensure that the purpose of the main tasks and how we use the annotations are clearly explained in the qualification task to the potential annotators.

Cost For reference agreement, a total of 1500 document-level annotations with 13401 phrase-level evaluations are collected from the qualified annotators, costing approximately $1400. For faithfulness, we collect 6450 phrase-level annotations for KP20k and 6486 annotations for KPTimes, costing approximately $800. We adjust the unit pay to ensure $15 hourly pay.

---

[13]We choose **M20** as many supervised models' outputs largely overlap with those included in the annotation process.

Document Title: extensional normalisation and type directed partial evaluation for typed lambda calculus with sums
Reference Keyphrases: normalisation, typed lambda calculus, grothendieck logical relations, strong sums
Predictions (M21): typed lambda calculus, sums, extensional normalisation, grothendieck logical relations

- -

Human:
$P = (1.00 + 0.67 + 0.60 + 1.00)/4 = 0.82$
$R = (0.73 + 1.00 + 1.00 + 0.67)/4 = 0.85$
$F1 = 2 \times 0.85 \times 0.82/(0.85 + 0.82) = 0.83$
Exact Matching: $P = 0.50$, $R = 0.50$, $F1 = 0.50$
Substring Matching: $P = 1.00$, $R = 1.00$, $F1 = 1.00$
BertScore: $P = 0.94$, $R = 0.96$, $F1 = 0.95$
Semantic Matching:
$P = (1.00 + 0.62 + 0.55 + 1.00)/4 = 0.79$
$R = (0.55 + 1.00 + 1.00 + 0.62)/4 = 0.79$
$F1 = 2 \times 0.79 \times 0.79/(0.79 + 0.79) = 0.79$

Figure 7: An example case from KP20k. Human scores are the average of the scores from three annotators, normalized to a [0,1] range. The small differences in human precision and recall scores are due to annotation noises. Semantic matching's intermediate and final scores are the most similar to human judgments.

Interface The annotation instructions and interface for reference agreement are presented in Figure 10 and Figure 11. The interface for faithfulness is presented in Figure 13.

E Qualitative Study

E.1 $SemF1$ vs. existing metrics

In Figure 7, we present a qualitative example where the model (**M21**) predicts two keyphrases exactly matching to some reference keyphrases and two keyphrases semantically similar to the reference. Correspondingly, human annotators assign partial credits to both of the "near-misses". However, for these phrases, exact matching gives a score 0 and substring matching gives full credit. As the reference and the prediction contain many similar tokens, BertScore is also near 1.0. Semantic matching's scoring is the most similar to humans.

E.2 Rethinking model comparisons

In Figure 8, we compare ExHiRD-h and CatSeqTG-2RF1 on two instances from KP20k. When evaluated with exact matching, CatSeqTG-2RF1 is preferred. However, since a lot of correct keyphrases are not recognized by exact matching, the irrelevant concepts and duplications are under-penalized by $F1@M$. By contrast, these issues are identified by the metrics from KPEVAL dedicated to faithfulness and diversity. With semantic-based evaluation, KPEVAL suggests that ExHiRD-h outperforms CatSeqTG-2RF1 on these two instances.

Title: yet another write optimized dbms layer for flash based solid state storage .

Abstract: flash based solid state storage (flashsss) has write oriented problems such as low write throughput , and limited life time . especially , flashssds have a characteristic vulnerable to random writes , due to its control logic utilizing parallelism between the flash memory chips . in this paper , we present a write optimized layer of dbmss to address the write oriented problems of flashsss in on line transaction processing environments . the layer consists of a write optimized buffer , a corresponding log space , and an in memory mapping table , closely associated with a novel logging scheme called incremental logging (icl) . the icl scheme enables dbmss to reduce page writes at the least expense of additional page reads , while replacing random writes into sequential writes . through experiments , our approach demonstrated up to an order of magnitude performance enhancement in i o processing time compared to the original dbms , increasing the longevity of flashsss by approximately a factor of two .

Reference: `icl ; ssd ; incremental logging ; flash memory ; write performance ; database`

ExHiRD-h: `solid state storage ; flash memory ; write optimized buffer ; incremental logging`
  – **Exact matching:** $F1@M = 0.500$ (Present), $F1@M = 0$ (Absent)
  – **KPEval:** $SemF1 = 0.657$, faithfulness= 0.864, $emb\_sim = 0.322$, $RR = 1.0$

CatSeqTG-2RF1: `flash based solid state storage ; flash based solid state storage ; incremental logging ; security ; flash memory ; flash memory ; flash memory`
  – **Exact matching:** $F1@M = 0.667$ (Present), $F1@M = 0$ (Absent)
  – **KPEval:** $SemF1 = 0.585$, faithfulness= 0.773, $emb\_sim = 0.419$, $RR = 0.333$

Title: bicepstrum based blind identification of the acoustic emission (ae) signal in precision turning .

Abstract: it is believed that the acoustic emissions (ae) signal contains potentially valuable information for monitoring precision cutting processes , as well as to be employed as a control feedback signal . however , ae stress waves produced in the cutting zone are distorted by the transmission path and the measurement systems . in this article , a bicepstrum based blind system identification technique is proposed as a valid tool for estimating both , transmission path and sensor impulse response . assumptions under which application of bicepstrum is valid are discussed and diamond turning experiments are presented , which demonstrate the feasibility of employing bicepstrum for ae blind identification .

Reference: `acoustic emissions ; higher order statistics ; blind identification ; precision machining`

ExHiRD-h: `acoustic emission ; precision turning ; blind system identification ; acoustic emission ; blind source separation`
  – **Exact matching:** $F1@M = 0.333$ (Present), $F1@M = 0$ (Absent)
  – **KPEval:** $SemF1 = 0.629$, faithfulness= 0.825, $emb\_sim = 0.122$, $RR = 1.0$

CatSeqTG-2RF1: `bicepstrum ; blind identification ; acoustic emission ; precision turning ; diamond turning ; algorithms ; blind source separation ; blind source separation ; blind source separation`
  – **Exact matching:** $F1@M = 0.4$ (Present), $F1@M = 0$ (Absent)
  – **KPEval:** $SemF1 = 0.556$, faithfulness= 0.532, $emb\_sim = 0.336$, $RR = 0.333$

Figure 8: A comparison between ExHiRD-h and CatSeqTG-2RF1 on two instances from KP20k. KPEVAL challenges the results of exact matching with more fine-grained evaluation signals.

KP20k:
For each paper, write a short citation text that summarizes some idea reflected in the abstract without copying anything here. Use a fake paper id like [3] or [5] to refer to the paper. Do not present in a summary format. Instead, write as if you are citing the paper in another paper.

Title: How Should I Explain? A Comparison of Different Explanation Types for Recommender Systems
Abstract: Recommender systems help users locate possible items of interest more quickly by filtering and ranking them in a personalized way. In particular, we present the results of a user study in which users of a recommender system were provided with different types of explanation. Our study reveals that the content-based tag cloud explanations are particularly helpful to increase the user-perceived level of transparency and to increase user satisfaction even though they demand higher cognitive effort from the user. Based on these insights and observations, we derive a set of possible guidelines for designing or selecting suitable explanations for recommender systems.
Citation: The ability for an artificially intelligent system to explain recommendations has been shown to be an important factor for user acceptance and satisfaction [13].

... two examples omitted ...

Title: [document_title]
Abstract: [document_abstract]
Citation:

- -

KPTimes:
For each piece of news, write several phrases as ad-hoc queries that some people might write if they want to find this article on the Internet. Write your response in 3-5 phrases and separate the phrases with commas.
Title: [document_title]
Abstract: [document_abstract]
Citation:

Figure 9: Prompts used for instructing GPT-4 to generate the ad-hoc queries for utility evaluation.

What do you need to do?

First, you will be given a list of reference phrases and a list of phrases to evaluate. We will also give a document from which these phrases are extracted. This document is only used to give you some context.

Title: security personalization for internet and web services .

First few sentences: the growth of the internet has been accompanied by the growth of internet services (e.g. , e commerce , e health) . this proliferation of services and the increasing attacks on them by malicious individuals have highlighted the need for service security .

References: security ; personalization ; web services ; internet services ; security policy ; negotiation

Phrases to Evaluate: security personalization ; security model ; security policy ; policy of securing ; model being

For each phrase in "**Phrases to Evaluate**", find a best-match (i.e., most similar) keyphrase from "**References**". Then, on a scale from 1 (not similar at all) to 5 (identical), tell us the semantic similarity between the two. Separate each entry with a comma:

Scores: 4, 3, 5, 4, 1

Explanation: we explain 5 phrases one-by-one. (1) "security personalization" matches best to either "security" or "personalization" and is pretty similar to them, so we assign a score 4. (2) "security model" may be matched best to "security", and they have a moderate semantic similarity, so we assign a score 3. (3) "security policy" exactly matches to the same phrase in the input, so we give a score 5. (4) "policy of securing" matches best to "security policy" and they have a considerable similarity, so we give a score 4. (5) "model being" is not similar at all with any phrase in the reference, so we give a score 1.

Figure 10: An example of the annotation instructions for the keyphrase evaluation study in §5.3.

Document title: a variant of parallel plane sweep algorithm for multicore systems .

First few sentences: parallel algorithms used in very large scale integration physical design bring significant challenges for their efficient and effective design and implementation . the rectangle intersection problem is a subset of the plane sweep problem , a topic of computational geometry and a component in design rule checking , parasitic resistance capacitance extraction , and mask processing flows . a variant of a plane sweep algorithm that is embarrassingly parallel and therefore easily scalable on multicore machines and clusters , while exceeding the best known parallel plane sweep algorithms on real world tests , is presented in this letter .

For each phrase in "Phrases to Evaluate", find a best-match (i.e., most similar) keyphrase from "References". Then, on a scale from 1 (not similar at all) to 5 (identical), tell us the semantic similarity between the two. You should give 1 only sparingly, i.e., when there are really nothing in the reference that semantically relates to the phrase evaluated. Separate each entry with a comma:

References: plane sweep, multicore, physical design, rectangle intersection, computational geometry

Phrases to Evaluate: parallel plane sweep, multicore systems, rectangle intersection, computational geometry, design rule checking, parallel algorithms

(Hint: you should write 6 scores below.)

Scores: [   ]

Figure 11: An example of the annotation interface for the keyphrase evaluation study in §5.3.

Title: high radix montgomery modular exponentiation on reconfigurable hardware .

Abstract: it is widely recognized that security issues will play a crucial role in the majority of future computer and communication systems . central tools for achieving system security are cryptographic algorithms . this contribution proposes arithmetic architectures which are optimized for modern field programmable gate arrays (fpgas) . the proposed architectures perform modular exponentiation with very long integers . this operation is at the heart of many practical public key algorithms such as rsa and discrete logarithm schemes . we combine a high radix montgomery modular multiplication algorithm with a new systolic array design . the designs are flexible , allowing any choice of operand and modulus . the new architecture also allows the use of high radices . unlike previous approaches , we systematically implement and compare several variants of our new architecture for different bit lengths . we provide absolute area and timing measures for each architecture . the results allow conclusions about the feasibility and time space trade offs of our architecture for implementation on commercially available fpgas . we found that 1,024 bit rsa decryption can be done in 3.1 ms with our fastest architecture .

Candidates: cryptography, discrete logarithm, exponentiation, field programmable gate arrays, fpga, gate arrays, hardware implementation, high radix, high radix modular multiplication, modular arithmetic, modular exponentiation, montgomery, montgomery modular multiplication, montgomery multiplication, public key cryptography, reconfigurable hardware, rsa, systolic array

Keyphrases (multiple keyphrases, separated by a comma):

[   ]

Figure 12: An example of the annotation interface for the keyphrase annotation study in Appendix C.5.

Please annotate

Document title: baby boomers ready to sell businesses to the next generation

First few sentences: baby boomers ready to sell businesses to the next generation dan bizzarro and bill jordan were still in college when they went to work at the bimac corporation , a castings manufacturer in dayton , ohio . after graduation , they joined full time as manufacturing and engineering executives . and eventually they teamed with a third employee to buy the business . they had a good ride . the company , with [digit] employees , has annual sales of about $ [digit] . [digit] million . but as the two men entered their 60s , after nearly three decades as owners , they began to wonder how long the arrangement could endure . their partner , roger reedy , had been talking for several years about retiring and moving to florida . after considering various options , including buying mr .

Keyphrase context: the tally of completed transactions is also rising , and the median sale price is up [digit] [kp-start] percent [kp-end] compared with prices a year ago .

Predicted Keyphrase: percent

Is this keyphrase faithful to the document? ○ Yes   ○ No

Figure 13: An example of the annotation interface for the faithfulness study in §6.1.