Match More, Extract Better! Hybrid Matching Model for Open Domain Web Keyphrase Extraction

Mingyang Song, Liping Jing*, Yi Feng
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, Beijing, China
mingyang.song@bjtu.edu.cn

Abstract

Keyphrase extraction aims to automatically extract salient phrases representing the critical information in the source document. Identifying salient phrases is challenging because there is a lot of noisy information in the document, leading to wrong extraction. To address this issue, in this paper, we propose a hybrid matching model for keyphrase extraction, which combines representation-focused and interaction-based matching modules into a unified framework for improving the performance of the keyphrase extraction task. Specifically, HybridMatch comprises (1) a PLM-based Siamese encoder component that represents both candidate phrases and documents, (2) an interaction-focused matching (IM) component that estimates word matches between candidate phrases and the corresponding document at the word level, and (3) a representation-focused matching (RM) component captures context-aware semantic relatedness of each candidate keyphrase at the phrase level. Extensive experimental results on the OpenKP dataset demonstrate that the performance of the proposed model HybridMatch outperforms the recent state-of-the-art keyphrase extraction baselines. Furthermore, we discuss the performance of large language models in keyphrase extraction based on recent studies and our experiments.

1 Introduction

Keyphrase Extraction (KE) is a fundamental task in natural language processing that aims to identify a series of salient and representative phrases relevant to the primary information discussed in a source document. Specifically, as shown in Table 1, keyphrases are concise content that encapsulates the central semantic meaning of a longer text (Hasan and Ng, 2014; Song et al., 2023b). Keyphrases play a crucial role in various natural language processing and information retrieval applications (Song et al., 2021a, 2022b, 2023h,a). Keyphrases can generally be categorized into two types: present keyphrases, which appear in the document, and absent keyphrases, which do not.

The Input Document:
Home of the Top Heavy Pizza Sandwiches Welcome *54 Pizza Express* offers a high quality *homemade product* served with friendly efficient service for Dinein Carry out or Delivery Every *54 Pizza Express* pizza is a quality pizza made with *wholesome quality ingredients* fresh vegetables delectable meats rich tangy tomato sauce and creamy mozzarella cheese Made fresh daily and served with your choice of dressing Choose from a Side Salad Dinner Salad Chef Salad Griled Chicken Salad Tuna Salad Buffalo Grilled Chicken Salad or Pepperoni and ... ... offer a high quality *homemade product* served with friendly efficient service for Dinein Carry out or Delivery Every *54 Pizza Express* pizza is a quality pizza made with *wholesome quality ingredient* fresh vegetables delectable meats rich tangy tomato sauce and creamy mozzarella cheese At *54 Pizza Express* we provide quality you can afford and a taste ...
Keyphrases: *54 Pizza Express*, *wholesome quality ingredient*, *homemade product*

Table 1: Sample the input document with keyphrases in the OpenKP dataset. For the input document, italic content indicates the keyphrases.

Keyphrase extraction models usually consist of two components: candidate keyphrase extraction, which selects candidate keyphrases from the given document by some heuristics (Liu et al., 2009; Grineva et al., 2009; Medelyan et al., 2009; Liang et al., 2021), and keyphrase importance estimation, which determines whether a candidate keyphrase is a true keyphrase via unsupervised (Mihalcea and Tarau, 2004; Liang et al., 2021) or supervised approaches (Mu et al., 2020; Sun et al., 2020; Song et al., 2021b). Most existing supervised keyphrase extraction approaches have focused on estimating the importance of keyphrases effectively and adequately, such as the recent studies (Sun et al., 2020; Wang et al., 2020; Song et al., 2021b).

* Corresponding Author

Findings of the Association for Computational Linguistics ACL 2024, pages 17–27

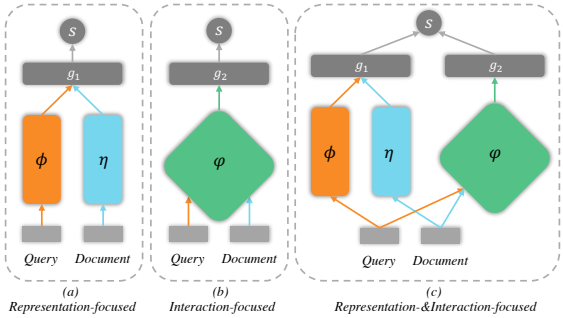

Figure 1: Illustration of three main categories of matching approaches: (a) Representation-focused matching models, (b) Interaction-focused matching models, and (c) Unified models, which integrate both representation- and interaction-focused matching techniques.

Although recent state-of-the-art keyphrase extraction approaches have made significant progress, they ignore explicitly modeling the relevance between candidate phrases and their corresponding document. Consequently, Song et al. (2022a) explicitly models the phrase-document relevance by using the Poincaré distance in the hyperbolic space, further improving the performance of keyphrase extraction, as presented in Figure 1 (a). However, a document often contains redundant information and typically has more complex semantics than each candidate phrase. Therefore, directly calculating the relevance between the entire candidate keyphrases and their corresponding document may lead to wrong keyphrase extraction.

Inspired by the issue above, we propose a new hybrid matching model, HybridMatch, to enhance the performance of the keyphrase extraction task. HybridMatch first evaluates the importance scores of candidate phrases from multiple granularities and then aggregates these scores into a final importance score for ranking and extracting keyphrases. Specifically, HybridMatch comprises a PLM-based Siamese encoder, an interaction-focused matching module(IM), and a representation-focused matching module (RM). The PLM-based Siamese encoder is employed to encode candidate phrases and their corresponding documents. The IM module estimates the phrase-document relevance at the word level, providing a surface-level importance score for each candidate keyphrase. Meanwhile, the RM module assesses the phrase-document similarity to capture the semantic relatedness at the phrase level, yielding high-level importance scores for the candidate keyphrases. Finally, we combine these two matching scores into a total score and utilize

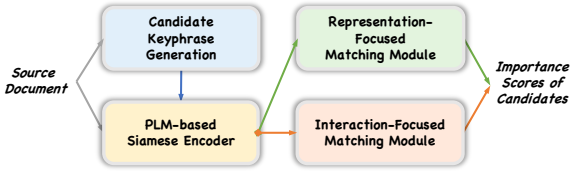

Figure 2: The overall architecture of HybridMatch.

a ranking model to train the entire system. Extensive experimental results on the OpenKP dataset demonstrate the effectiveness of our model by comparing with recent state-of-the-art baselines. To summarize, our contributions are fourfold:

- A hybrid matching model is proposed to estimate the importance of candidate keyphrases from different perspectives by taking advantage of both interaction- and representation-focused matching models.
- An interaction-focused matching module estimates the relevance scores between candidate phrases and their corresponding document at the word level.
- A representation-focused matching module captures context-aware semantic relatedness between candidate keyphrases and their corresponding document.
- Extensive experiments on the OpenKP dataset show that HybridMatch outperforms the recent state-of-the-art keyphrase extraction baselines. An ablation test is performed to demonstrate the superiority of each component.

The rest of this paper is organized into six Sections. Among them, Section 2 describes the proposed HybridMatch model. The experimental setting and results are discussed in Section 3 and Section 4. Section 5 discusses the related work. A brief conclusion is given in Section 6.

2 Methodology

In this section, we describe the architecture of HybridMatch. Figure 2 shows the proposed model HybridMatch, consisting of three components: a PLM-based Siamese encoder, an interaction-focused matching module, and a representation-focused matching module. In the following sections, the details of HybridMatch are given.

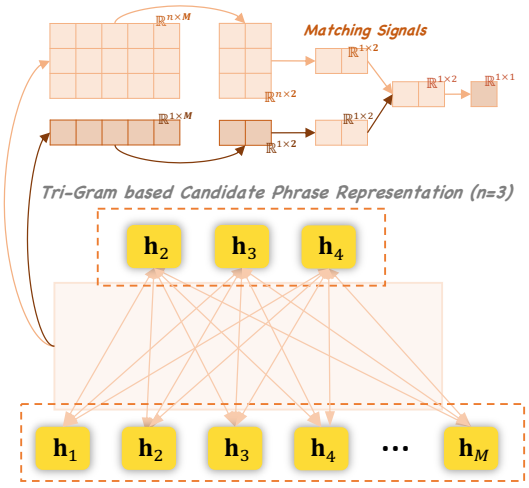

Figure 3: The hybrid matching module.

2.1 Problem Definition

Traditional keyphrase extraction typically can be formulated as follows: given the source document D, the keyphrase extraction model aims at obtaining a keyphrase set P. In this paper, however, we treat the keyphrase extraction task as a text matching problem, which can be reformulated as follows: given a candidate keyphrase p and its corresponding document D, the text matching model aims to compute and learn a matching score $f(D, p)$ using the candidate phrase terms $\{x_1^p, ..., x_n^p\}$ and document terms $\{x_1^D, ..., x_M^D\}$, where n and M are the number of terms in p and D respectively. To be clear, "term" refers to either a single word or aggregating all words in phrase p or document D.

2.2 Siamese Encoder

In this section, we describe the contextualized text encoder. To minimize information loss, we represent phrases and documents at varying levels of granularity.

2.2.1 Document Representation

To embed the given document D, we adopt the pre-trained language model (RoBERTa (Liu et al., 2019)) as the backbone to obtain word representations $\mathbf{H} = \{\mathbf{h}_{cls}, \mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_m, ..., \mathbf{h}_M\}$,

$$\mathbf{H} = \text{RoBERTa}(D), \tag{1}$$

where $\mathbf{h}_m \in \mathbb{R}^d$ is the word representation of the m-th word x_m in the document D. Here, $\mathbf{h}_{cls}$ is the representation of the [CLS] token of the pre-trained language model RoBERTa, which indicates the whole document representation.

2.2.2 Candidate Keyphrase Representation

As previously mentioned, we treat keyphrase extraction as a matching task involving creating a set of candidate keyphrases from the given document, similar to the two-stage keyphrase extraction methods (Sun et al., 2020; Song et al., 2021b). In this section, we provide details on how we construct the candidate keyphrase set for the given document.

Based on the RoBERTa, we can obtain the output representation $\mathbf{H}$. Concretely, to allow for a better collection of candidate keyphrase representations, we only retain tokens corresponding to each word in the source document for word-level document representation $\mathbf{H}$, thus removing some special symbols (i.e., [CLS] and paddings). After the above operations, we can obtain each word representation $\hat{\mathbf{H}} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_M\} \in \mathbb{R}^{M \times d}$ of the given document, and M indicates the actual length of the document D. Next, our model composes words to n-grams, where $\mathbf{h}^n_{m:m+n}$ indicates the m-th n-gram candidate phrase representation (contains n words). Specifically, we use $\mathbf{h}^n_{m:m+n}$ as the word-level phrase representation. Furthermore, we also aggregate the word-level phrase representation to a whole phrase representation as follows,

$$\mathbf{h}^n_m = \text{CNN}^n(\mathbf{h}^n_{m:m+n})\mathbf{W}_c, \tag{2}$$

where $\mathbf{W}_c$ is the learnable matrix and CNN^n indicates the Convolutional Neural Network with the kernel size of $n \in [1, N]$. Here, N denotes the maximum length of the extracted candidate keyphrases.

2.3 Interaction-focused Matching Module

Since keyphrases are frequently used to retrieve documents in various real-world applications, a robust keyphrase extraction system should comprehensively model and leverage the relevance of candidate keyphrases to their respective documents. However, documents often contain redundant information. To address this, we propose an interaction-focused matching module that captures matching signals at the word level, as illustrated in Figure 3. Specifically, we calculate the relevance score between each candidate keyphrase and the corresponding document to obtain the word-level phrase-document relevance scores.

To obtain the word-level phrase-document relevance scores, we calculate the textual similarity between candidate phrases and their corresponding document by multiplying the word-level candidate

phrase representation $\mathbf{h}_m^n \in \mathbb{R}^{n\times d}$ and the word-level document representation $\hat{\mathbf{H}} \in \mathbb{R}^{M\times d}$,

$$\mathbf{S}^{pp} = \mathbf{h}_{m:m+n}^n \hat{\mathbf{H}}^\top, \mathbf{S}^{pp} \in \mathbb{R}^{n\times M}, \quad (3)$$

$$\mathbf{S}_{i,j}^{pp} = \text{softmax}(\mathbf{S}_{i,j}^{pp}) = \frac{e^{\mathbf{S}_{i,j}^{pp}}}{\sum_{k=1}^{m} e^{\mathbf{S}_{i,k}^{pp}}}, \quad (4)$$

where $\mathbf{S}_{i,j}^{pp}$ can be considered the word-level similarity score by matching the i-th word representation of the candidate keyphrase with the j-th word representation of the document. Specifically, since the candidate keyphrase and the document are embedded in the same pre-trained language model (RoBERTa), similar words will be placed closer in the semantic space, and their product will produce larger scores.

Then, we apply *max*, *mean* pooling to the similarity matrix to obtain discriminative feature vectors:

$$\text{Max}(\mathbf{S}^{pp}) = [\text{Max}(\mathbf{S}_{1,:}^{pp}), ..., \text{Max}(\mathbf{S}_{n,:}^{pp})], \quad (5)$$

$$\text{Mean}(\mathbf{S}^{pp}) = [\text{Mean}(\mathbf{S}_{1,:}^{pp}), ..., \text{Mean}(\mathbf{S}_{n,:}^{pp})], \quad (6)$$

where $\text{Max}(\mathbf{S}^{pp}) \in \mathbb{R}^n$, $\text{Mean}(\mathbf{S}^{pp}) \in \mathbb{R}^n$ indicate the matching signals of each candidate keyphrases. Each score generated from pooling can be viewed as one piece of matching evidence for a specific query term or phrase to the document, and its value denotes the importance of the relevance signal. Compared to *max* pooling, *mean* pooling is beneficial for cases where a phrase is matched to multiple relevant terms in the document as follows,

$$r^{pp} = \text{Mean}([\text{Max}(\mathbf{S}^{pp}); \text{Mean}(\mathbf{S}^{pp})]), \quad (7)$$

where $[;]$ indicates the concatenation of two vectors and $r^{pp} \in \mathbb{R}^{1\times 2}$ indicates the word-level phrase-document relevance matching signals. In addition, when two keyphrases have similar semantics, the keyphrase with more words often gets a higher score. Therefore, the word-level matching module may alleviate such problems.

2.4 Representation-focused Matching Module

The word-level phrase-document relevance matching typically focuses on the local view of salient information in each candidate keyphrase, often overlooking valuable content that could be used for extracting keyphrases from the document. In contrast, representation-focused matching emphasizes "meaning" correspondences by leveraging linguistic information (such as words, phrases, and entities) and compositional structures (like dependency trees). Therefore, in this paper, we first introduce a representation-focused matching (RM) approach to address these limitations, which captures semantic similarities through attention mechanisms applied to candidate phrases and their corresponding document representations.

Hyperparameter	Dimension or Value
RoBERTa Embedding ($\mathbb{R}^d$)	768
Max Sequence Length	512
Maximum Phrase Length (N)	5
λ_1 and λ_2	0.5
δ	1.0
Optimizer	AdamW
Batch Size	64
Learning Rate	5×10^{-5}
Warm-Up Proportion	10%

Table 2: Parameters used for training HybridMatch.

To obtain the phrase-level phrase-document relevance matching scores, we calculate the textual similarity between the candidate phrase and the document by multiplying the phrase-level candidate phrase representation $\mathbf{h}_m^n \in \mathbb{R}^{n\times d}$ and the word-level document representation $\hat{\mathbf{H}} \in \mathbb{R}^{M\times d}$,

$$\mathbf{S}^{pw} = \mathbf{h}_m^n \hat{\mathbf{H}}^\top, \mathbf{S}^{pw} \in \mathbb{R}^{1\times M}, \quad (8)$$

$$r^{pw} = \text{softmax}([\text{Max}(\mathbf{S}^{pw}); \text{Mean}(\mathbf{S}^{pw})]), \quad (9)$$

where $\mathbf{S}^{pw}$ represents the phrase-level relevance of the whole candidate keyphrase to all words in the document and $r^{pw} \in \mathbb{R}^{1\times 2}$ denotes the phrase-level phrase-document relevance matching scores.

2.5 Matching Scores Aggregation

Usually, the importance of words in a sentence depends on the importance of the sentence as a whole (Hsu et al., 2018). Similarly, the importance of words in a phrase depends on whether the phrase as a whole is essential. Therefore, we further influence the word-level phrase-document relevance by using the phrase-level phrase-document relevance as a weighting factor as the relevance matching score as follows,

$$r_m^n = \text{Mean}(r^{pp} \odot r^{pw}), \quad (10)$$

where $r_m^n \in \mathbb{R}^{1\times 1}$ denotes the final score and $\odot$ indicates an element-wise product. This multiplica-

tion ensures that the updated word-level relevance score can be high only when both word-level and phrase-level relevance scores are high. Furthermore, this weighting method reduces the impact of high matching scores from common words like stopwords. Our relevance matching has two essential properties. First, all the operations, including matching, softmax, pooling, and weights, have no learnable parameters. Second, the parameter-free nature makes our model more interpretable and robust from over-fitting. Finally, the importance score of each candidate keyphrase can be calculated as follows,

$$f(D,p) = \lambda_1 \text{FNN}_h(\mathbf{h}_m^n) + \lambda_2 r_m^n, \quad (11)$$

where λ_1 and λ_2 are used to balance the influence of two matching modules, $\text{FNN}_h(\cdot)$ is a 2-layer feed-forward network, and $f(D,p)$ indicates the relevance matching score and will be used for ranking in the training procedure.

2.6 Training and Inference

To train the whole model, the pairwise hinge loss is adopted for training and optimizing the model parameters, which is widely used in the field of NLP and IR, formulated as,

$$\mathcal{L}_{rank} = \max(0, \delta - f(D,p^+) + f(D,p^-)), \quad (12)$$

where $\mathcal{L}_{rank}$ is the pairwise loss based on a triplet of the document D, a relevant (positive) keyphrase sample p^+, and an irrelevant (negative) keyphrase sample p^-. Following the previous studies (Sun et al., 2020; Song et al., 2021b, 2022a, 2023h), we use the same approach to select positive and negative samples from all candidates.

3 Experimental Settings

In this section, we first introduce the used dataset. Then, we present the evaluation metric, baselines, and implementation details.

3.1 Datasets

The OpenKP[1] dataset is an open-domain keyphrase extraction dataset encompassing a wide range of domains and topics. It includes over 150,000 real-world online pages, each annotated with the most relevant key terms by experts. In this paper, we utilize the OpenKP dataset as our primary benchmark, adhering to its official splits: 134,000 documents for training, 6,600 documents for development, and 6,600 documents for testing.

[1] https://github.com/microsoft/OpenKP

3.2 Evaluation Metrics

Following previous studies (Xiong et al., 2019; Sun et al., 2020; Song et al., 2021b, 2023h), Precision (P), Recall (R), and F-measure (F1) of the top K keyphrase predictions are used for evaluating the performance of keyphrase extraction approaches. Specifically, we adopt $K = 1, 3, 5$ on the OpenKP dataset.

Furthermore, we use Porter Stemmer[2] (Porter, 2006) to determine the match of two phrases, which is consistent with prior studies (Meng et al., 2017; Xiong et al., 2019).

3.3 Implementation and Training Details

We adopt the pre-trained language model RoBERTa as the backbone of our model, initialized from their pre-trained weights. Our models are optimized using AdamW (Li and Li, 2018) with a 5e-5 learning rate and 10% warm-up proportion. Here, we set the batch size to 64 and the maximum sequence length to 512. Following the previous work (Sun et al., 2020; Song et al., 2021b), the maximum phrase length is set to 5 (N = 5). The training process of our model uses eight RTX A4000 GPUs. In addition, our code is implemented based on PyTorch 1.8[3]. Table 2 shows the details.

3.4 Baselines

We compare HybridMatch with two types of models to comprehensively prove the effectiveness of our models. We first compare traditional unsupervised methods TF-IDF (Jones, 2004) and TextRank (Mihalcea and Tarau, 2004). Then, we compare three kinds of keyphrase extraction methods that utilize variants of the pre-trained language models as the backbone to extract keyphrases by spanning (Sun et al., 2020), chunking (Sun et al., 2020), ranking (Sun et al., 2020), multi-task learning(Song et al., 2021b), and hyperbolic deep learning (Song et al., 2022a). There are also no additional features available in many real applications, such as visual features. Meanwhile, neither the above selection baselines nor the proposed model utilizes additional feature information to assist in estimating the importance of candidate phrases, so BLING-KPE (Xiong et al., 2019) and SMART-KPE+R2J (Wang et al., 2020) are not selected as baselines.

[2] https://tartarus.org/martin/PorterStemmer/
[3] https://pytorch.org/

Model	OpenKP								
	P@1	P@3	P@5	R@1	R@3	R@5	F1@1	F1@3	F1@5
TFIDF	28.3	18.4	13.7	15.0	28.4	34.7	19.6*	22.3*	19.6*
TextRank	7.7	6.2	5.5	4.1	9.8	14.2	5.4*	7.6*	7.9*
SpanKPE-BERT†	48.1	28.4	20.8	26.1	43.6	52.0	32.4	33.1	28.8
ChunkKPE-BERT	51.1	30.6	22.5	27.1	46.4	55.8	34.0	35.6	31.1
RankKPE-BERT	51.3	32.3	23.5	27.3	48.9	58.2	34.2	37.4	32.5
HybridMatch-BERT	52.4	33.3	23.9	28.3	49.7	60.2	35.4	38.2	33.1
SpanKPE-RoBERTa†	51.8	31.0	22.6	27.8	47.4	56.6	34.7	36.1	31.3
ChunkKPE-RoBERTa	53.3	32.2	23.5	28.3	48.6	58.1	35.5	37.3	32.4
RankKPE-RoBERTa	53.8	33.7	24.4	29.0	50.9	60.4	36.1	39.0	33.7
HybridMatch-RoBERTa	54.6	34.2	24.9	30.4	52.5	62.1	37.3	39.4	34.1

Table 3: Model performances on the OpenKP dataset. *F1@3 is the main metric for this dataset* (Xiong et al., 2019; Wang et al., 2020). The best results are underlined. * denotes these results not included in the original paper and are estimated with precision and recall scores. † indicates that these results are evaluated via the code provided by its corresponding paper. The results of the baselines are reported in their corresponding papers.

4 Results and Analysis

In this section, we show the results of the baselines and our proposed models. We also show the analyses of HybridMatch in different scenarios, including an ablation test and a case study. In addition, we also discuss the performance of large language models in keyphrase extraction based on recent studies and our experiments.

4.1 Overall Performance

Our primary experimental results on the OpenKP dataset are presented in Table 3, and the best results are underlined. Specifically, for the OpenKP dataset, F1@3 is the primary evaluation metric[4]. As shown in Table 3, we can see that our model HybridMatch achieves better performance on all evaluation metrics than the previous state-of-the-art baselines. This proves that considering both interaction-focused and representation-focused matching can extract keyphrases more accurately. Meanwhile, the proposed models beat other ranking-based competitive baselines (RankKPE-BERT and RankKPE-RoBERTa) by a large margin and are still superior to those baselines (SpanKPE-BERT, SpanKPE-RoBERTa, ChunkKPE-BERT, and ChunkKPE-RoBERTa). Furthermore, this finding further suggests that filtering the redundant information and selecting the saliency from multiple views is beneficial for keyphrase extraction.

[4] https://microsoft.github.io/msmarco/

Model	OpenKP		
	F@1	F@3	F@5
HybridMatch	**37.3**	**39.4**	**34.1**
HybridMatch-IM	37.1	38.2	33.0
HybridMatch-RM	36.9	38.9	33.9

Table 4: Ablation tests on the OpenKP dataset. The best results are highlighted in bold.

4.2 Ablation Test

To better understand the contribution of each module in our proposed model, we conducted an ablation study on the base HybridMatch model by systematically removing each component. We aimed to examine how the interaction-focused matching and representation-focused matching modules contribute to the model's effectiveness. The results on the OpenKP dataset are presented in Table 4, with each row indicating the removal of a specific module. For instance, the row labeled "HybridMatch-IM" represents the model with the representation-focused matching module removed.

The results demonstrate that each component of our model (HybridMatch-IM and HybridMatch-RM) outperforms the compared baselines. This is because, unlike previous baselines, our model considers multiple perspectives to select important keyphrases from the document.

The Input Document: Dining Services Dining Overview Meal plans Eating on campus The dining services website has details about dining plans for students living on campus and for commuters places to eat Dining Choices food options including vegetarian vegan and foods free of the top 8 allergens dining hours and more Requirement Meal plans are required for students living on campus excluding the apartments View details below Sign up for a meal plan online through the same system as for oncampus housing How to make changes... You can change your meal plan before movein day at the same signup link After movein and through the first 14 days of the semester you can change your meal plan by contacting Jean Hoyt in Nazareth Campus Operations ... or stop by her office in Smyth 60 Meal plan requirements Resident students living on campus excluding students living in Breen Lyons and Portka apartments are required to choose a resident dining meal plan Each meal plan also comes with bonus dollars to their Nazareth ID card throughout ... (URL: https://www2.naz.edu/dining-overview/) **Target Keyphrase**: (1) ***nazareth dining*** ; (2) ***meal plans*** ; (3) ***resident students***
HybridMatch: (1) ***meal plans***; (2) dining services; (3) dining plans; (4) ***resident students***; (5) eating on campus **HybridMatch-RM**: (1) ***meal plans***; (2) dining services; dining plans; (3) eating on (4) campus; (5) dining overview **HybridMatch-IM**: (1) ***meal plans***; (2) dining services; (3) eating on campus; (4) meal plan; (5) campus

Table 5: Example of keyphrase extraction results on the OpenKP dataset (contains partial of the input document and target keyphrases). Phrases in orange and bold are keyphrases predicted by the different models.

	PROMPTS
TP1	`Extract keywords from this text: [Document]`
TP2	`Extract keyphrases from this text: [Document]`
TP_{Zero}	`You are good at extracting keyphrases from the given document. Keyphrases reflect the core content and topic information of the document. Please extract keyphrases from the given document and follow the format [“”,“”,“”,“”,“”] to return the answer. Do not give any explanation. [Document]`

Table 6: Three prompts are designed for extracting keyphrases from the document, where TP1 and TP2 are proposed by Song et al. (2023c). TP_{Zero} is the prompt designed in this paper.

Moreover, combining the two kinds of matching modules significantly enhances the performance of keyphrase extraction. This improvement is attributed to integrating different matching signals, which leads to a more accurate estimation of the importance scores of each candidate keyphrase.

4.3 Case Study

In this section, we show an example from the OpenKP dataset in Figure 5. We compare the extraction results of our three models, including HybridMatch, HybridMatch-IM, and HybridMatch-RM. Concretely, the output keyphrases extracted by the HybridMatch-IM model are more inclined to extract the content that appears several times in the given document or the content that appears more frequently. For the output keyphrases extracted by the HybridMatch-RM model, it is more inclined to extract keyphrases that match the semantics of the given document. Overall, the HybridMatch model combines the advantages of both IM and RM modules to focus on both semantically important and frequently occurring keyphrases and ignore the extraction of interfering items such as stop words. This example shows that the joint modeling of semantic and relevance matching can improve the performance of keyphrase extraction.

4.4 LLMs for Keyphrase Extraction

Large Language Models (LLMs) have achieved good results in many natural language processing downstream tasks. However, in previous studies (Song et al., 2023c,e), getting good results in keyphrase extraction or generation tasks is not easy

```
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant
to the user question displayed below. Your evaluation should consider factors such as the helpfulness,
relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by
providing a short explanation. Be as objective as possible. After providing your explanation, please rate
the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating:
[[5]]".
#Question
Extract keyphrases from the document
#Document
{document}
#Keyphrases
{keyphrases}
```

Table 7: Prompt used for evaluating the OpenKP dataset. Concretely, {document} indicates the input document and {keyphrases} denotes the output keyphrases.

Model	F@1	F@3	F@5
ChatGLM (TP2)†	11.5	8.5	7.1
ChatGLM2 (TP2)†	16.0	11.0	8.6
GPT-3.5-Turbo (TP1)†	16.5	21.1	17.4
GPT-3.5-Turbo (TP2)†	18.0	21.6	18.7
GPT-4-Turbo (TP_{Zero})	21.8	23.5	21.0
HybridMatch	**37.3**	**39.4**	**34.1**

Table 8: Results of our model and zero-shot test of large language models on the OpenKP dataset. † indicates that these results are given from Song et al. (2023c). GPT-4-Turbo-128K (OpenAI, 2024) indicates the *gpt4-1106-preview* model with a 128K context window.

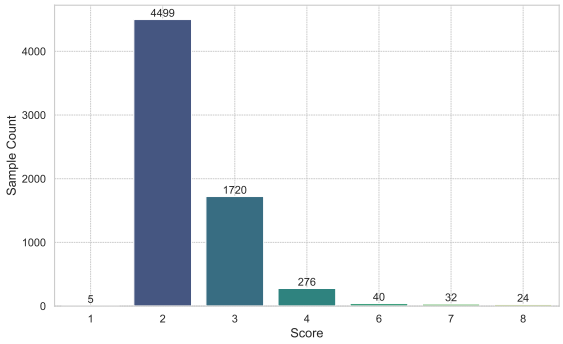

Figure 4: Evaluation results.

for LLMs, which may be inferior to small models trained on specific domain data. Therefore, in this paper, we also test the performance of LLMs on the keyphrase extraction task, i.e., GPT-4-Turbo. Unlike previous work (Song et al., 2023c), we enrich and improve the prompt for extracting keyphrases, as shown in Table 6. The results show that even GPT-4-Turbo does not meet expectations in the keyphrase extraction task.

Based on the issue mentioned earlier, we suppose that the most likely reason is that the annotation results in the existing keyphrase extraction or generation datasets are unreliable or of low quality. Therefore, referring to Zheng et al. (2023), we adopt GPT-4-Turbo to evaluate the annotation information in the validation set of the OpenKP dataset via the prompt shown in Table 7. The evaluation results are shown in Figure 4. Indeed, the results align with our hypothesis. The scores evaluated by GPT-4-Turbo mostly fall between 2 and 3, indicating potential issues with the annotations of the OpenKP dataset. We also selected three unsupervised keyphrase extraction datasets (e.g., DUC2001 (Wan and Xiao, 2008b), Inspec (Hulth, 2003), SemEval2010 (Kim et al., 2010)) for testing the performance of GPT-4-Turbo and compared them with a SOTA prompt-based keyphrase extraction method (i.e., PromptRank (Kong et al., 2023)). From the results in Table 9, we can see that there are still many differences in the effects of GPT-4-Turbo on different keyphrase extraction benchmarks. Therefore, in the future, unlocking the potential of LLMs to improve the performance of the keyphrase extraction task, utilizing LLMs to build a new benchmark for the keyphrase extraction and generation tasks, or correcting the annotations of existing benchmarks using some leading LLMs may be valuable research directions.

5 Related Work

Typically, keyphrase extraction aims to automatically extract a set of phrases related to the primary information discussed in the source document (Tomokiyo and Hurst, 2003; Hasan and Ng, 2014; Meng et al., 2017; Song et al., 2023b). Specifically, recent keyphrase extraction baselines can be mainly

divided into two components: candidate keyphrase extraction and keyphrase importance estimation. The former uses heuristics to select a candidate keyphrase set (Grineva et al., 2009; Mihalcea and Tarau, 2004; Song et al., 2022c; Witten et al., 1999; Wan and Xiao, 2008a; Medelyan et al., 2009; Song et al., 2023f,d,g,i) for the given document, and the latter is to measure the importance of candidates for extracting keyphrases (Sun et al., 2019; Zhang et al., 2020; Sun et al., 2020; Wang et al., 2020; Song et al., 2021b, 2022a, 2023h).

Unlike the existing methods, in this paper, we combine interaction-focused and representation-focused matching into a unified matching model for extracting keyphrases.

6 Conclusion

In this paper, we propose a new hybrid matching model (HybridMatch) to combine representation-focused and interaction-focused matching modules into a unified model. To filter the noisy information, we first design an interaction-focused matching module and a representation-focused matching module to estimate the important scores of candidate phrases from different granularities. Then, we aggregate the relevance matching scores from IM and RM as the importance scores to rank and extract keyphrases. Extensive experiments demonstrate the effectiveness of our model, and ablation studies verify the performance of each component. In addition, we discuss the performance of large language models in keyphrase extraction based on recent studies and our experiments. In the future, it will be interesting to introduce a graph-based network to capture the interactions from multiple granularities to calculate the phrase-document relevance as the importance of each candidate for improving the performance of keyphrase extraction.

7 Limitations

There are still some limitations to our work. In this paper, we follow previous research and generate candidate keyphrases using the n-gram rule, which may introduce many noisy or irrelevant keyphrases. Therefore, improving the procedure for extracting candidate keyphrases may be interesting and valuable in improving the upper bound of keyphrase extraction performance. One possible approach is to prune candidate keyphrases from the source document by utilizing surface-level information of candidate keyphrases.

Model	DUC2001		
	F@5	F@10	F@15
PromptRank	**27.4**	**31.6**	**31.0**
GPT-3.5-Turbo (TP2)†	21.5	25.0	24.21
GPT-4-Turbo (TP_{Zero})	26.5	27.1	25.0
Model	Inspec		
	F@5	F@10	F@15
PromptRank	31.7	37.9	38.2
GPT-3.5-Turbo (TP2)†	28.1	34.9	36.7
GPT-4-Turbo (TP_{Zero})	**35.0**	**40.3**	**40.7**
Model	SemEval2010		
	F@5	F@10	F@15
PromptRank	17.2	20.7	21.4
GPT-3.5-Turbo (TP2)†	13.7	16.1	16.2
GPT-4-Turbo (TP_{Zero})	**18.4**	**22.7**	**23.6**

Table 9: Keyphrase extraction results on the DUC2001, Inspec, and SemEval2010 datasets.

8 Acknowledgments

We thank the three anonymous reviewers for carefully reading our paper and their insightful comments and suggestions.

This work was partly supported by the National Natural Science Foundation of China under Grant 62176020; the Joint Foundation of the Ministry of Education for Innovation team (8091B042235); the Beijing Natural Science Foundation under Grant L211016; the Fundamental Research Funds for the Central Universities (2019JBZ110); and the State Key Laboratory of Rail Traffic Control and Safety (Contract No. RCS2023K006), Beijing Jiaotong University.

References

Maria P. Grineva, Maxim N. Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *WWW*, pages 661–670. ACM.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *ACL (1)*, pages 1262–1273. The Association for Computer Linguistics.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia. Association for Computational Linguistics.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP*.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *SemEval@ACL*, pages 21–26. The Association for Computer Linguistics.

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.

Zhize Li and Jian Li. 2018. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *NeurIPS*, pages 5569–5579.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Unsupervised keyphrase extraction by jointly modeling local and global context. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 155–164, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *EMNLP*, pages 257–266. ACL.

O. Medelyan, E. Frank, and I. H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Internat. Conference of Empirical Methods in Natural Language Processing, EMNLP-2009,*.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *ACL*, pages 582–592. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*, pages 404–411. ACL.

Funan Mu, Zhenting Yu, Lifeng Wang, Yequan Wang, Qingyu Yin, Yibo Sun, Liqun Liu, Teng Ma, Jing Tang, and Xing Zhou. 2020. Keyphrase extraction with span-based feature representations. *CoRR*, abs/2002.05407.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

M.F. Porter. 2006. An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 40(3):211–218.

Mingyang Song, Yi Feng, and Liping Jing. 2022a. 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5710–5720. Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2022b. A preliminary exploration of extractive multi-document summarization in hyperbolic space. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 4505–4509. ACM.

Mingyang Song, Yi Feng, and Liping Jing. 2022c. Utilizing BERT intermediate layers for unsupervised keyphrase extraction. In *5th International Conference on Natural Language and Speech Processing, ICNLSP 2022, Trento, Italy, December 16-17, 2022*, pages 277–281. Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2023a. Hisum: Hyperbolic interaction model for extractive multi-document summarization. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 1427–1436. ACM.

Mingyang Song, Yi Feng, and Liping Jing. 2023b. A survey on recent advances in keyphrase extraction from pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2108–2119. Association for Computational Linguistics.

Mingyang Song, Xuelian Geng, Songfang Yao, Shilong Lu, Yi Feng, and Liping Jing. 2023c. Large language models as zero-shot keyphrase extractors: A preliminary empirical study. *CoRR*, abs/2312.15156.

Mingyang Song, Haiyun Jiang, Lemao Liu, Shuming Shi, and Liping Jing. 2023d. Unsupervised keyphrase extraction by learning neural keyphrase set function. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2482–2494. Association for Computational Linguistics.

Mingyang Song, Haiyun Jiang, Shuming Shi, Songfang Yao, Shilong Lu, Yi Feng, Huafeng Liu, and Liping Jing. 2023e. Is chatgpt A good keyphrase generator? A preliminary study. *CoRR*, abs/2303.13001.

Mingyang Song, Liping Jing, Yi Feng, Zhiwei Sun, and Lin Xiao. 2021a. Hybrid summarization with semantic weighting reward and latent structure detector. In *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 1739–1754. PMLR.

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

Mingyang Song, Huafeng Liu, Yi Feng, and Liping Jing. 2023f. Improving embedding-based unsupervised keyphrase extraction by incorporating structural information. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1041–1048. Association for Computational Linguistics.

Mingyang Song, Huafeng Liu, and Liping Jing. 2023g. HyperRank: Hyperbolic ranking model for unsupervised keyphrase extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16070–16080, Singapore. Association for Computational Linguistics.

Mingyang Song, Lin Xiao, and Liping Jing. 2023h. Learning to extract from multiple perspectives for neural keyphrase extraction. *Comput. Speech Lang.*, 81:101502.

Mingyang Song, Pengyu Xu, Yi Feng, Huafeng Liu, and Liping Jing. 2023i. Mitigating over-generation for unsupervised keyphrase extraction with heterogeneous centrality detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16349–16359, Singapore. Association for Computational Linguistics.

Si Sun, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. 2020. Joint keyphrase chunking and salience ranking with bert. *CoRR*, abs/2004.13639.

Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. Divgraphpointer: A graph pointer network for extracting diverse keyphrases. In *SIGIR*, pages 755–764. ACM.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. pages 33–40. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008a. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *COLING*, pages 969–976.

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

Yansen Wang, Zhen Fan, and Carolyn Penstein Rosé. 2020. Incorporating multimodal information in open-domain web keyphrase extraction. In *EMNLP (1)*, pages 1790–1800. 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 *ACM DL*, pages 254–255. ACM.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *EMNLP/IJCNLP (1)*, pages 5174–5183. Association for Computational Linguistics.

Haoyu Zhang, Dingkun Long, Guangwei Xu, Pengjun Xie, Fei Huang, and Ji Wang. 2020. Keyphrase extraction with dynamic graph convolutional networks and diversified inference. *CoRR*, abs/2010.12828.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.