Retrieval-Augmented Multilingual Keyphrase Generation with
Retriever-Generator Iterative Training∗

Yifan Gao1, Qingyu Yin1,‡, Zheng Li1,‡, Rui Meng2, Tong Zhao1, Bing Yin1, Irwin King3, Michael R. Lyu3
1Amazon Inc. 2Salesforce Research 3Chinese University of Hong Kong
1{yifangao,qingyy,amzzhe}@amazon.com 2ruimeng@salesforce.com

Abstract

Keyphrase generation is the task of automatically predicting keyphrases given a piece of long text. Despite its recent flourishing, keyphrase generation on non-English languages hasn’t been vastly investigated. In this paper, we call attention to a new setting named multilingual keyphrase generation and we contribute two new datasets, EcommerceMKP and AcademicMKP, covering six languages. Technically, we propose a retrieval-augmented method for multilingual keyphrase generation to mitigate the data shortage problem in non-English languages. The retrieval-augmented model leverages keyphrase annotations in English datasets to facilitate generating keyphrases in low-resource languages. Given a non-English passage, a cross-lingual dense passage retrieval module finds relevant English passages. Then the associated English keyphrases serve as external knowledge for keyphrase generation in the current language. Moreover, we develop a retriever-generator iterative training algorithm to mine pseudo parallel passage pairs to strengthen the cross-lingual passage retriever. Comprehensive experiments and ablations show that the proposed approach outperforms all baselines.†

1 Introduction

Keyphrases are single or multi-word lexical units that best summarize a piece of text. As such, they are of great importance for indexing, categorizing, and mining in many information retrieval and natural language processing tasks (Jones and Staveley, 1999; Frank et al., 1999; Hulth and Megyesi, 2006; Dave et al., 2003). Keyphrase generation is the task of automatically predicting keyphrases given a piece of long text. Existing works on keyphrase generation mostly focus on English datasets (Gallina et al., 2019; Meng et al., 2017) while keyphrase generation for languages other than English is still under-explored. Since search engines usually provide services to customers using different languages, multilingual keyphrase generation becomes a significant problem while it is still unknown how well existing keyphrase generation approaches perform in non-English languages.

Nevertheless, there are two challenges we will face regarding multilingual keyphrase generation. First, to the best of our knowledge, there is no large-scale dataset publicly available for training and benchmarking multilingual keyphrase generation models. Building keyphrase datasets at a sufficient scale is difficult and costly. Second, compared with the existing datasets in English, which can contain millions of data examples and cover a wide diversity of topics, the data resources in non-English languages are inherently scarce. For example, in the domain of e-commerce, marketplaces using English have abundant customer queries to be used for keyphrase mining, while queries in some languages are relatively less than in English, which is probably because of a smaller size of user population or a shorter operation time.

We start tackling these challenges by contributing two new datasets for multilingual keyphrase, which cover six languages and two domains. The first dataset EcommerceMKP is collected from a real-world major e-commerce website. The product descriptions are used as the source text while the target keyphrases are manually provided. Comprehensive experiments and ablations show that the proposed approach significantly outperforms all baselines.‡

† Qingyu Yin and Zheng Li are corresponding authors. This work was mainly done while Rui Meng was a Ph.D. student at University of Pittsburgh.
‡ The datasets are released at https://github.com/Yifan-Gao/multilingual_keyphrase_generation.
ademic papers in Chinese and Korean are included in AcademicMKP.

To overcome the resource scarcity challenge in training multilingual models, we propose a retrieval-based method to leverage the keyphrase knowledge in large-scale English datasets. By investigating multilingual keyphrase data, we observe that data in different languages may talk about similar topics. Therefore, we conjecture that passage-keyphrases pairs in English can be of help as an external knowledge base for multilingual keyphrase generation. To be specific, given a passage in low-resource language XX, we propose to use a retrieval model to find multiple top-related passages in English. These retrieved English passages provide high-quality English keyphrases that can be used as hints for generating keyphrases in other languages. After that, the generator takes the code-mixed inputs, including the passage in language XX and retrieved English keyphrases, and predicts keyphrases in language XX.

In the cross-lingual retrieval training, parallel passage-keyphrases pairs between English and other languages are extremely limited. For example, in the e-commerce domain, only a small fraction of products have both English and non-English descriptions (being sold in multiple countries). Such a data scarcity issue weakens the ability of cross-lingual knowledge acquisition from high-resource English keyphrases as intermediary, and finally hinders the potential of the retrieval-augmented keyphrase generation. To mitigate the problem, we propose a retriever-generator iterative training (RGIT) algorithm to automatically mine pseudo training parallel pairs from unlabeled data. Concretely, the retriever can dynamically adjust in terms of the current variations of generation performance between the proposed retrieval-augmented generator and the base one without the aid of retrieved English keyphrases. Starting from insufficient seed parallel pairs, if the retrieved pseudo passage-keyphrases pairs in the current iteration can bring in higher generation results as the generator’s feedback, those pseudo parallel data will be regarded as high quality and incorporated into the seed ones to further boost the retriever. Such cycle providing positive effects can be repeated until the increasing generation performance stopped.

We conduct extensive experiments on EcommerceMKP and AcademicMKP and demonstrate that large-scale English datasets do provide useful knowledge for multilingual keyphrase generation. The proposed retrieval-augmented method outperforms traditional extraction-based models, sequence-to-sequence neural models, and its variants. Moreover, the RGIT algorithm boosts the retrieval performance significantly by mining over 20k pseudo-parallel passage pairs. We also conduct detailed analyses to investigate the effectiveness of retriever-generator iterative training.

2 Related Work

Keyphrase Generation. The advance of neural language generation enables models to freely generate keyphrases according to the phrase importance and semantics, rather than extracting a list of sub-strings from the text (Witten et al., 1999a; Liu et al., 2011; Wang et al., 2016). Meng et al. (2017) propose the first keyphrase generation model Copy-RNN, which not only generates words based on a vocabulary but also points to words in the source text — overcoming the barrier of predicting absent keyphrases. Following this idea, Chen et al. (2018); Zhao and Zhang (2019); Ahmad et al. (2021) leverage the attention mechanism to reduce duplication and improve coverage. Chen et al. (2019b); Ye and Wang (2018); Wang et al. (2019); Liang et al. (2021) propose to leverage extra structure information (e.g., title, topic) to guide the generation. Chan et al. (2019); Luo et al. (2021) propose a model using reinforcement learning, and Swaminathan et al. (2020) propose using GAN for KPG. Chen et al. (2020) introduce hierarchical decoding and exclusion mechanism to prevent models from generating duplicate phrases. Ye et al. (2021b) propose to dynamically align target phrases to eliminate the influence of order, as highlighted by Meng et al. (2021). Mu et al. (2020); Liu et al. (2020a); Park and Caragea (2020) use pre-trained language models for better representations of documents.

Retrieval Augmented Text Generation (RAG) recently shows great power in knowledge-intensive NLP tasks such as open-domain question answering, fact checking and entity linking (Lewis et al., 2020; Petroni et al., 2021; Guu et al., 2020). In RAG, a retriever (either sparse (Lee et al., 2019) or dense (Karpukhin et al., 2020)) searches for useful non-parametric knowledge from a knowledge base, then a generator combines the non-parametric retrieved knowledge with its parametric knowledge, learned during pre-training, for solving the task. Different from these tasks, keyphrase gen-
Cross-Lingual Dense Passage Retrieval

External Knowledge
(English Passage-Keyphrases Pairs)

Multilingual Keyphrase Generation with Code-Mixed Inputs

External Knowledge
(English Passage-Keyphrases Pairs)

Passage \( p^{xx} \) → Retriever → Top m EN Passages
\( p^{EN}_1, p^{EN}_2, ..., p^{EN}_m \) → Generator → Keyphrases
\( k^{ES}_1, k^{ES}_2, ..., k^{ES}_n \)

Figure 1: Overview of our Retrieval-Augmented Multilingual Keyphrase Generation (RAMKG) framework. \( p^{xx}, k^{xx} \) denote a passage and keyphrases in language \( XX \) (\( XX \in \{ DE, ES, FR, IT, KO, ZH \} \)). \( p^{EN}, k^{EN} \) denote relevant passages and keyphrases retrieved from the English dataset.

eration is not a knowledge-intensive task but we treat the English passage-keyphrase training data as our knowledge. Similar approaches have been investigated in neural machine translation (Gu et al., 2018; Cai et al., 2021), dialogue (Weston et al., 2018), and knowledge-base QA (Das et al., 2021). In keyphrase generation, Chen et al. (2019a); Ye et al. (2021a); Kim et al. (2021) retrieve similar documents from training data to produce more accurate keyphrases. However, their retrieval module is a non-parametric model and cannot be generalized in the multilingual setting due to the large vocabulary gap between languages.

3 Task Definition

In this paper, we aim to tackle the keyphrase generation task in a multilingual setting, which means one model of desire is capable of generating keyphrases in any language that it has been trained with. The benefits of having a single keyphrase generation model for multiple languages are threefold: (1) Collecting keyphrase annotation for individual language can be prohibitively expensive; (2) Training and deploying separate models for each language is laborious; (3) Joint training of multiple languages can alleviate the resource scarcity by utilizing rich monolingual data.

Formally, we define the multilingual keyphrase generation task as follows. Given a piece of text \( p^{xx} \) in language \( XX \), our goal is to predict its corresponding keyphrases \( k^{xx}_1, k^{xx}_2, ..., k^{xx}_n \) in language \( XX \), where \( n \) is the total number of target keyphrases for this text \( p^{xx} \). In this study, \( XX \) can be German (DE), Spanish (ES), Italian (IT), French (FR), Korean (KO) or Chinese (ZH).

4 Model

Scarcity of resources is one of the topmost challenges for multilingual tasks, which is also the case for multilingual keyphrase generation. One may find it difficult to collect enough text data in languages other than English, much less the annotation of keyphrases in specific domains. To overcome this problem, we propose a retrieval-augmented approach to make use of the relatively rich resources in English. The motivation for our proposed retrieval-augmented approach comes from an observation from data: texts and keyphrases expressed in different languages usually share common topics or knowledge concepts. For example, in e-commerce websites, it is often the case that the same products are sold in different marketplaces/countries. Thus these products as well as their keyphrases, though exhibited in different languages, can share a high semantic similarity. In other words, given a text in language \( p^{AA} \), if we could find a similar text in language \( p^{BB} \), its associated keyphrases \( k^{BB} \) may serve as a good hint for the to-be-generated keyphrases \( k^{AA} \) in language \( AA \). Since English has the most abundant text-keyphrases pairs in both e-commerce and academic papers domains, its resource can be treated as a non-parametric keyphrase knowledge base, which provides texts in English covering a wide range of topics and concepts, as well as the associated high-quality keyphrases.

As shown in Fig. 1, our framework consists of a retrieval step and a generation step:

1. Retrieval Step: given a source passage \( p^{xx} \) in language \( XX \), the cross-lingual retriever first finds \( m \) semantically relevant English passages \( p^{EN}_1, p^{EN}_2, ..., p^{EN}_m \). Each retrieved En-
English passage $p_{j}^{EN}$ has its associated $n_j$ English keyphrases $k_{j,1}^{EN}, k_{j,2}^{EN}, ..., k_{j,n_j}^{EN}$. These retrieved English keyphrases are taken as external knowledge for keyphrase generation in step 2.

2. **Generation Step**: taking the source text $p_{i}^{XX}$ in language $XX$ and all retrieved English keyphrases $\{k_{i,1}^{EN}, ..., k_{i,n}^{EN}\}$, as inputs, the generation module concatenates them as a sequence and generates keyphrases in target language $XX$.

### 4.1 Cross-Lingual Dense Passage Retrieval

The cross-lingual retriever includes a passage encoder $E_P(\cdot)$ and a query encoder $E_Q(\cdot)$. The passage encoder $E_P(\cdot)$ maps millions of English passages into $d$-dimensional vectors and builds indices for all English passages using FAISS (Johnson et al., 2021) offline. At inference time, the passage in language $XX$ goes through the query encoder $E_Q(\cdot)$ and is converted into a $d$-dimensional vector. Then the cross-lingual retriever performs a KNN search to retrieve $m$ English passages whose vectors are closest to the query vector measured by the dot product similarity: $\text{sim}(p_{i}^{XX}, p_{j}^{EN}) = E_Q(p_{i}^{XX})^T E_P(p_{j}^{EN})$.

**Passage Encoder.** Since naive lexical similarity can hardly handle text matching across languages, we resort to a BERT-based dense passage retriever (Karpukhin et al., 2020), expecting the contextualized semantic matching can retrieve similar passages accurately and robustly. In order to meet the demand of multilingual representation, we utilize multilingual pre-trained model mBERT (Devlin et al., 2019) to encode passages into 768-dimensional vectors.

**Training.** Since the output vectors of mBERT are not aligned across languages, we need extra alignment training to ensure that similar passages in different languages can be mapped into near regions in the high-dimensional space. Given a passage $p_{i}^{XX}$ in language $XX$, we take its corresponding English passage $p_{i}^{EN}$ as the positive example and randomly select $n$ negative passages $p_{i,n_1}^{EN}, ..., p_{i,n}^{EN}$ in the English corpus. The dense retriever is trained by optimizing the negative log likelihood loss of the positive English passage.

In the e-commerce domain, we select the positive passage according to product metadata. For a product sold in both EN and XX marketplaces, we regard its bilingual product descriptions $(p_{i}^{XX}, p_{i}^{EN})$ as a parallel passage pair, i.e., positive training example. For the domain of academic paper, we notice that papers with parallel text is very rare. Therefore, we develop an automatic approach to mine parallel abstract pairs of English and the target language. Specifically, we adopt an off-the-shelf bi-text mining tool named Sentence Transformers (Reimers and Gurevych, 2019) to mine pseudo parallel pairs. Given two datasets in different languages, we encode passages using LaBSE (Feng et al., 2020), the current best method for learning language-agnostic sentence embeddings for 109 languages, and then parallel passages can be extracted through nearest-neighbor retrieval and filtered by setting a fixed threshold over a margin-based similarity score, as proposed in (Artetxe and Schwenk, 2019).

### 4.2 Multilingual Keyphrase Generation with Code-Mixed Inputs

Given the top $m$ retrieved English passages $p_{1}^{EN}, ..., p_{m}^{EN}$, we find their associated keyphrases in the
dataset: \{k_{1,1}^{EN}, \ldots, k_{m,1}^{EN}\}, \ldots, \{k_{1,n_1}^{EN}, \ldots, k_{m,n_m}^{EN}\}.

We utilize mBART (Liu et al., 2020b), a multilingual denoising pre-trained sequence-to-sequence language model, to integrate information from multiple languages. Different from machine translation which maps a sentence a the source language to a target language, our multilingual keyphrase generation model takes code-mixed inputs – a combination of retrieved English keyphrases \{k_{1,1}^{EN}, \ldots, k_{m,1}^{EN}\}, \ldots, \{k_{1,n_1}^{EN}, \ldots, k_{m,n_m}^{EN}\} from m retrieved English passages and the source passage \(p_{XX}\) in the target language \(XX\).

We concatenate retrieved English keyphrases with a delimiter token [SEP], and add special tokens to separate different inputs: [ENKPS] for retrieved keyphrases and [CTX] for the source passage. Besides, we follow the fine-tuning setup of mBART by adding the language identifier [XX] (e.g. [DE] for German) at the end of the input sequence to denote the current input language:

\[\text{[ENKPS]} k_{1,1}^{EN} \text{[SEP]} \ldots \text{[SEP]} k_{m,n_m}^{EN} \text{[CTX]} p_{XX} \text{[XX]}\].

The training task is a sequence of concatenated keyphrases \(k_{1,XX}, \ldots, k_{n,XX}\), separated by a special token [SEP]. [XX], the language identifier of the current language, is also added at the beginning of the target sequence to indicate the target language:

\[\text{[XX]} k_{1,XX} \text{[SEP]} k_{2,XX} \text{[SEP]} \ldots \text{[SEP]} k_{n,XX}\].

### 4.3 Retriever-Generator Iterative Training

In spite of having utilized parallel passage pairs to align the multilingual representations of the retrieval module, it remains a concern because the parallel passage pairs between English and non-English languages account for only a small portion of the whole multilingual dataset. For example, in a popular e-commerce platform, only a small percentage of products (less than 10%) have both English and non-English descriptions. Without enough quality parallel pairs, the cross-lingual dense passage retriever may not work well to find relevant English passages. Consequently, associated English keyphrases may provide little help for multilingual keyphrase generation.

To make the multilingual keyphrase generation generalize better to any target languages or domains without reliance on numerous parallel passage pairs, we propose an iterative training method to mine parallel passages which requires only a small number of initial parallel pairs of bootstrap the process. Since our ultimate goal of the retriever model is to provide useful external knowledge for multilingual keyphrase generation, we mine parallel passage pairs (English and a non-English language) according to whether the retrieved English passage-keyphrases pairs could help the keyphrase generation for the target non-English language \(XX\). For example, let \(p_{EN}^0\) and \(p_{EN}^0\) be two retrieved English passages for a passage \(p_{XX}\) in target language, if the associated keyphrases of \(p_{EN}^0\) provide more useful information for generating the keyphrases of \(p_{XX}\) than \(p_{EN}^1\), then \(p_{EN}^0, p_{EN}^0\) would be considered as a better parallel passage pair. That said, we expect the mined pseudo parallel passage pairs to be of high quality according to the retrieval score, at the same time they can be directly helpful for training the generation module.

The proposed iterative training approach is sketched in Algo. 1 and Fig. 2. Given a large-scale keyphrase dataset in English \(D_{LS} = \{(p_{LS}^{EN}, k_{LS}^{EN})\}\) and a smaller one \(D_{XX} = \{(p_{XX}^{EN}, k_{XX}^{EN})\}\) in target language \(XX\), we denote the set of annotated parallel examples (bilingual passages in English and other languages) as \(\text{PARallel split} D_{PAR} = \{(p_{PAR}^{EN}, p_{PAR}^{XX}, k_{PAR}^{EN}, k_{PAR}^{XX})\}\), in which \(\{(p_{PAR}^{EN}, k_{PAR}^{EN})\}\) comes from the XX dataset while \(\{(p_{PAR}^{XX}, k_{PAR}^{XX})\}\) comes from the English dataset. The remaining data examples in the target dataset

---

**Algorithm 1** Parallel Passage Mining via Iterative Training

1. **Input:** (1) Parallel data \(D_{PAR} = \{(p_{PAR}^{EN}, p_{PAR}^{XX}, k_{PAR}^{EN}, k_{PAR}^{XX})\}\). (2) Non-parallel data \(D_{NP} = \{(p_{NP}^{EN}, k_{NP}^{EN})\}\). (3) Large-scale English corpus \(D_{LS} = \{(p_{LS}^{EN}, k_{LS}^{EN})\}\).

2. **Output:** Pseudo parallel passage pairs \(D_{PSEUDO}\)

3. \(\triangleright 0. \) Train Seq2Seq baseline w/o retrieved keyphrases

4. \(G_B \leftarrow \text{train}((\{(p_{LS}^{EN}, k_{LS}^{EN})\}) \in D_{PAR})\)

5. \(D_{PSEUDO} \leftarrow \{\}\) // Pseudo parallel passage pairs

6. **for** \(t \in \{0 \ldots T - 1\} \) **do**

7. \(\triangleright 1. \) Train retriever on pseudo and parallel data

8. \(R_{\text{LS}} \leftarrow \text{train}((\{(p_{LS}^{EN}, k_{LS}^{EN})\}) \in D_{PSEUDO} \cup D_{PAR})\)

9. \(\triangleright 2. \) Train retrieval-augmented generator on \(D_{PAR}\)

10. **for** \((p_{LS}^{EN}, k_{LS}^{EN}) \in D_{PAR} \) **do**

11. \(p_{LS}^{\hat{EN}} \leftarrow R_{\text{LS}}(p_{LS}^{EN}, D_{LS}) \) // Retrieve EN passages

12. \(\{k_{LS}^{\hat{EN}}\} \leftarrow \hat{p}_{LS} \) // Find associated EN keyphrases

13. // Train retrieval-augmented generator with EN keyphrases

14. \(G_{\text{LS}} \leftarrow \text{train}((\{(p_{LS}^{EN}, k_{LS}^{EN})\}) \in D_{PAR}, \{k_{LS}^{EN}\})\)

15. \(\triangleright 3. \) Create pseudo parallel passage pairs

16. \(D_{PSEUDO}^{t+1} \leftarrow \{\}\)

17. **for** \(p_{LS}^{EN} \in D_{LS} \) **do**

18. \(p_{LS}^{\hat{EN}} \leftarrow R_{\text{LS}}(p_{LS}^{EN}, D_{LS}) \) // Retrieve EN passages

19. \(\{k_{LS}^{\hat{EN}}\} \leftarrow \hat{p}_{LS} \) // Find associated EN keyphrases

20. // Predict keyphrases w/o and w/ EN keyphrases

21. \(\{k_{LS}^{\hat{EN}}, G_{\text{LS}}\} \leftarrow G_{\text{LS}}(p_{LS}^{EN})\)

22. \(\{k_{LS}^{\hat{EN}}, G_{\text{LS}}\} \leftarrow G_{\text{LS}}(p_{LS}^{EN})\)

23. // If adding EN keyphrases leads to better keyword predictions, the retrieved EN keyphrases are taken as positive examples

24. \(\text{if } F(\{(k_{LS}^{\hat{EN}}, G_{\text{LS}})\}) - F_1(\{(k_{LS}^{\hat{EN}}, G_{\text{LS}})\}) > \tau \) **then**

25. \(D_{PSEUDO}^{t+1} \leftarrow D_{PSEUDO}^{t+1} \cup \{(p_{LS}^{EN}, p_{LS}^{EN})\}\)

26. return \(D_{PSEUDO}^{t+1}\)
Table 1: AcademicMKP & EcommerceMKP Dataset

<table>
<thead>
<tr>
<th>Language</th>
<th>Train Size</th>
<th>Dev Size</th>
<th>Test Size</th>
<th>Passage Length (Avg/Std/Mid)</th>
<th>#Keyphrases (Avg/Std/Mid)</th>
<th>Absent Kps %</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AcademicMKP Dataset</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese (ZH)</td>
<td>1,110</td>
<td>158</td>
<td>319</td>
<td>217/348/207</td>
<td>5/15</td>
<td>27.2%</td>
</tr>
<tr>
<td>Korean (KO)</td>
<td>774</td>
<td>110</td>
<td>222</td>
<td>115/301/111</td>
<td>4/10</td>
<td>37.7%</td>
</tr>
<tr>
<td>Total</td>
<td>1,884</td>
<td>268</td>
<td>541</td>
<td>171/371/155</td>
<td>4/14</td>
<td>31.3%</td>
</tr>
<tr>
<td><strong>EcommerceMKP Dataset</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>German (DE)</td>
<td>23,997</td>
<td>1,411</td>
<td>2,825</td>
<td>157/399/141</td>
<td>10/58</td>
<td>57.1%</td>
</tr>
<tr>
<td>Spanish (ES)</td>
<td>12,222</td>
<td>718</td>
<td>1,440</td>
<td>159/341/139</td>
<td>9/57</td>
<td>54.6%</td>
</tr>
<tr>
<td>French (FR)</td>
<td>16,966</td>
<td>998</td>
<td>2,000</td>
<td>163/341/144</td>
<td>9/58</td>
<td>63.0%</td>
</tr>
<tr>
<td>Italian (IT)</td>
<td>9,163</td>
<td>538</td>
<td>1,081</td>
<td>167/341/152</td>
<td>8/37</td>
<td>42.6%</td>
</tr>
<tr>
<td>Total</td>
<td>62,368</td>
<td>3,665</td>
<td>7,346</td>
<td>161/348/143</td>
<td>9/57</td>
<td>56.4%</td>
</tr>
</tbody>
</table>

$D_{XX}$ have no annotated corresponding English examples in $D_{LS}$ (the pairs may exist but are not known yet), and we name this set as the Non-Parallel split $D_{NP} = \{(p_{EN,k_{EN}}^{XX})\}$. We first fine-tune a mBART using only keyphrases data of target language $\{(p_{PAR,k_{PAR}}^{XX})\}$ in $D_{PAR}$ (Line 4). Then we start a loop to mine pseudo parallel passage pairs for refining the passage retriever. Each iteration is expected to bring in a higher quality of pseudo passage pairs, resulting in a better performance of retriever, with three steps:

1. We train a retriever $R_i$ using existing available EN-XX passage pairs $\{(p_{PAR,k_{PAR}}^{XX})\}$ from both parallel passage data $D_{PAR}$ and most up-to-date pseudo passage data $D_{PSEUDO}$ (Line 8).

2. We train a retrieval-augmented model $G_t$ using multilingual passages $\{(p_{EN,k_{EN}}^{XX})\}$ from the parallel data $D_{PAR}$ and retrieved English keyphrases $\{k_{LS}^{EN}\}$ from the English dataset $D_{LS}$ (Line 14). To get the retrieved English keyphrases for each passage $p_{PAR}$, we take the retriever $R_i$ trained in step (1) to do a KNN search for passages $p_{EN}^{EN}$ in $D_{LS}$ and find their associated keyphrases $\{k_{LS}^{EN}\}$ (Line 10-12).

3. For each passage $p_{EN}^{EN}$ in the non-aligned dataset $D_{NP}$, we also retrieve English passages $p_{EN}^{EN}$ and keyphrases $\{k_{LS}^{EN}\}$ from $D_{LS}$ (Line 18-19). Then the retrieved English passage $p_{EN}^{EN}$ will be taken as the parallel text to $p_{SP}^{XX}$ if its associated keyphrases $\{k_{LS}^{EN}\}$ provide useful information. The usefulness is measured by the keyphrase generation performance (F-score) between the retrieval-augmented generation model $G_t$ and the base model $G_0$ that does not use EN keyphrases (Line 21-25).

After $T$ iterations, we train the retriever on the pseudo data $D_{PSEUDO}^{T}$ and fine-tune it on the parallel data $D_{PAR}$. Then we treat it as our final retriever and train the generation model in Sec. 4.2.

5 Datasets

**EcommerceMKP Dataset** is collected from a popular E-commerce shopping platform. There are four languages we consider for building EcommerceMKP: German (DE), French (FR), Spanish (ES) and Italian (IT). The title, product description, and bullet description provided by manufacturers are concatenated and treated as source input. The keyphrases of each product are selected from search queries under the following protocol. First, given a product, we only keep search queries that lead to purchases and treat them as effective queries. Then phrases are chunked from these effective queries using AutoPhrase (Shang et al., 2018) and further ranked by their frequency. Our assumption is: the more times a phrase appears in effective search queries of a product, the more important a phrase is. Finally a threshold is set to filter out unimportant phrases. Under this protocol, we receive 73k examples over four languages. The statistics are shown in Table 1.

We collect the passages and keyphrases under the same protocol for the English (EN) dataset and name it as EcommerceMKP-EN. In total the English dataset contains 3 million passage-keyphrases pairs. To obtain the parallel passage pairs for training the cross-lingual dense passage retriever, we pair the product descriptions in different languages according to the product identification information. We select a total of 1,247 parallel passages from EcommerceMKP training set which include 480 passages for DE-EN, 244 for ES-EN, 340 for FR-EN, and 183 for IT-EN. Besides, we keep 1,000 parallel passage pairs in the DEV set of EcommerceMKP to evaluate the performance of retrieval and bi-text mining.

**AcademicMKP Dataset** is collected from the academic domain. We take the title and abstract of each paper as the source text and author-provided keywords as the target output. All papers are sampled from Microsoft Academic Graph (Sinha et al., 2015), a web-scale academic entity graph that contains multiple types of scholarly entities and relationships: field of study, author, institution, abstract, venue, and keywords. We use Spacy (https://spacy.io/) to detect the language of abstracts and keyphrases, and choose two languages Chinese (ZH) and Korean (KO) to construct the AcademicMKP dataset. Since Microsoft Academic Graph (MAG) is automatically crawled and
Table 2: Main results on the EcommerceMKP dataset. The best results are in bold. (RGIT: Iterative-training)

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>KP-Miner</td>
<td>27.87</td>
<td>28.00</td>
<td>32.31</td>
<td>30.92</td>
<td>31.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YAKE</td>
<td>32.48</td>
<td>32.27</td>
<td>31.85</td>
<td>31.50</td>
<td>31.23</td>
<td>27.30</td>
<td>27.85</td>
<td>26.01</td>
<td>29.57</td>
<td>28.05</td>
<td>26.73</td>
<td></td>
</tr>
<tr>
<td>mBART (multilingual)</td>
<td>32.72</td>
<td>29.77</td>
<td>28.16</td>
<td>29.50</td>
<td>28.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBART + Pretrain</td>
<td>33.40</td>
<td>32.60</td>
<td>32.45</td>
<td>28.17</td>
<td>27.60</td>
<td>31.51</td>
<td>30.02</td>
<td>30.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAMKG (ours)</td>
<td>34.38</td>
<td>33.05</td>
<td>32.15</td>
<td>29.33</td>
<td>27.77</td>
<td>28.00</td>
<td>32.31</td>
<td>30.92</td>
<td>31.44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBART (monolingual)</td>
<td>32.52</td>
<td>31.50</td>
<td>31.50</td>
<td>24.97</td>
<td>26.66</td>
<td>24.96</td>
<td></td>
<td></td>
<td></td>
<td>26.38</td>
<td>24.85</td>
<td>24.87</td>
</tr>
<tr>
<td>mBART (multi)</td>
<td>32.48</td>
<td>32.27</td>
<td>31.85</td>
<td>27.03</td>
<td>26.64</td>
<td>30.25</td>
<td>30.08</td>
<td>29.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBART + Joint</td>
<td>31.10</td>
<td>30.58</td>
<td>30.23</td>
<td>27.30</td>
<td>26.01</td>
<td>29.57</td>
<td>28.05</td>
<td>26.73</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAMKG (ours)</td>
<td>34.40</td>
<td>32.60</td>
<td>32.45</td>
<td>28.17</td>
<td>27.60</td>
<td>31.51</td>
<td>30.02</td>
<td>30.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAMKG + RGIT (ours)</td>
<td>34.38</td>
<td>33.05</td>
<td>32.15</td>
<td>29.33</td>
<td>27.77</td>
<td>28.00</td>
<td>32.31</td>
<td>30.92</td>
<td>31.44</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Results on AcademicMKP (mono: monolingual, multi: multilingual, Joint: EN Joint Train, Pretrain: EN Pretrain, RGIT: Iterative-training).

<table>
<thead>
<tr>
<th>Model</th>
<th>Chinese (ZH) P</th>
<th>Chinese (ZH) R</th>
<th>Chinese (ZH) F1</th>
<th>Korean (KO) P</th>
<th>Korean (KO) R</th>
<th>Korean (KO) F1</th>
<th>Average P</th>
<th>Average R</th>
<th>Average F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>YAKE</td>
<td>32.48</td>
<td>32.27</td>
<td>31.85</td>
<td>27.03</td>
<td>26.64</td>
<td>30.25</td>
<td>30.08</td>
<td>29.65</td>
<td></td>
</tr>
<tr>
<td>mBART (multi)</td>
<td>31.10</td>
<td>30.58</td>
<td>30.23</td>
<td>27.30</td>
<td>26.01</td>
<td>29.57</td>
<td>28.05</td>
<td>26.73</td>
<td></td>
</tr>
<tr>
<td>mBART + Joint</td>
<td>33.40</td>
<td>32.60</td>
<td>32.45</td>
<td>28.17</td>
<td>27.60</td>
<td>31.51</td>
<td>30.02</td>
<td>30.49</td>
<td></td>
</tr>
<tr>
<td>RAMKG (ours)</td>
<td>34.38</td>
<td>33.05</td>
<td>32.15</td>
<td>29.33</td>
<td>27.77</td>
<td>28.00</td>
<td>32.31</td>
<td>30.92</td>
<td>31.44</td>
</tr>
<tr>
<td>RAMKG + RGIT (ours)</td>
<td>34.38</td>
<td>33.05</td>
<td>32.15</td>
<td>29.33</td>
<td>27.77</td>
<td>28.00</td>
<td>32.31</td>
<td>30.92</td>
<td>31.44</td>
</tr>
</tbody>
</table>

Table 3: Results on AcademicMKP (mono: monolingual, multi: multilingual, Joint: EN Joint Train, Pretrain: EN Pretrain, RGIT: Iterative-training).

6 Experimental Setup

6.1 Evaluation Metrics

Keyphrase Generation. Let the ground truth keyphrases be $Y = \{k_1, k_2, ..., k_M\}$ and the predicted keyphrases be $\hat{Y} = \{\hat{k}_1, \hat{k}_2, ..., \hat{k}_M\}$, we compute the precision ($P(M)$), recall ($R(M)$) and F-score ($F(M)$) between $Y$ and $\hat{Y}$ as $P(M) = \frac{|Y \cap \hat{Y}|}{|Y|}$, $R(M) = \frac{|Y \cap \hat{Y}|}{|\hat{Y}|}$, $F(M) = \frac{2 \times P(M) \times R(M)}{P(M) + R(M)}$, where $|Y|$ denotes the number of keyphrases in the gold set $Y$. We only consider exact match of two keyphrases (with some post-processing such as lower-case) for $|Y \cap \hat{Y}|$. Then the average are computed for all languages in the test set.

Passage Retrieval. The quantity of retrieved English passages directly influences how much external knowledge could be utilized for keyphrase generation. Therefore, we evaluate the top-k recall (k=1,2,5,10,20) on the DEV set for evaluating retrieval performance.

6.2 Baselines and Ablations

We consider following baselines and ablations:

2) Unsupervised Graph-based Keyphrase Extraction: TextRank (Mihalcea and Tarau, 2004), TopicalPageRank (Sterckx et al., 2015), PositionRank (Florescu and Caragea, 2017), MultipartiteRank (Boudin, 2018).
3) Supervised Feature-based Keyphrase Extraction: KEA (Witten et al., 1999).
4) Neural-based Supervised Keyphrase Generation: CopyRNN, Transformer, Kea.
Supervised Keyphrase Generation: CopyRNN, Transformer; 5) mBART (monolingual): separately trained 6 mBART models on each language; 6) mBART (multilingual): a single mBART model on all languages; 7) mBART + EN Joint Train: a mBART model jointly trained on the multilingual data and English data (KP20K (Meng et al., 2017) for AcademicMKP; EcommerceMKP-EN for EcommerceMKP). 8) mBART + EN Pretrain: a mBART firstly pre-trained on the English data and then fine-tuned on the multilingual data. 9) RAMKG (Ours): The Retrieval-Augmented Multilingual Keyphrase Generation model (Sec. 4.1 & 4.2). 10) RAMKG + RGIT (Ours): RAMKG improved with retriever-generator iterative training (RGIT) (Sec. 4.3).

7 Results and Analyses

7.1 Main Results

Main results are shown in Table 2 & 3 for EcommerceMKP and AcademicMKP respectively, and we make the following observations:

- The unsupervised approaches, both statistical-based and graph-based, have robust results across all languages. PositionRank performs the best among all unsupervised approaches.

- The supervised approaches consistently outperform unsupervised approaches. The feature-based approach KEA receives a high recall by predicting more keyphrases while the CopyRNN receives a high precision. Different from the results on English keyphrase generation where Transformer and CopyRNN are comparable, the Transformer beats the CopyRNN by a large margin in the multilingual scenario.

- We observe that jointly training on all languages (mBART multilingual) receives better results than separately training on each language (mBART monolingual). This implies the ability of locating and summarizing key information is transferable across languages.

- Comparing different approaches using external large-scale English data, we find that our proposed RAMKG outperforms both “EN Joint Train” and “EN Pretrain”. This is because the retrieval-augmented approach provides auxiliary knowledge information as part of the input to the generation module, while the other two variants have to “infuse” the knowledge learned from English data to model parameters. Moreover, “EN Joint Train” and “EN Pretrain” have no positive effect on AcademicMKP dataset (Table 3). Compared with multilingual and English data are from the same website, there is still a domain gap between papers (multilingual) in AcademicMKP and papers (English) in KP20K.

- The retriever-generator self-training (RAMKG + Iter) alleviates the data scarcity issue with the help of stronger retriever: since the retriever can find more relevant English keyphrases, it leads to a general improvement on keyphrase performance across languages.

7.2 Effect of Iterative Training

Retrieval Results We investigate the effect of retriever-generator iterative training by comparing the retrieval recall for models trained w/o and w/ the mined pseudo parallel passage pairs. Results on the DEV set of EcommerceMKP are shown in the Table 4. With additional mined pseudo parallel passage pairs, the retriever improves the Recall@5 from 50.1% to 72.4%. And therefore, the better retrieved English keyphrases lead to a better generation performance (44.50 vs. 45.73 in Table 2).

Quantity and Quality of Pseudo Parallel Passages We show the quantity and quality of mined pseudo parallel passages in Table 5. After each iteration of passage mining, our algorithm can consistently find around 20k passage pairs from EcommerceMKP training set, which are nearly 20 times of the initial data. To assess the quality of mined passage pairs, we examine the label accuracy using the 1,000 parallel passage pairs from EcommerceMKP training set, which are nearly 20 times of the initial data. To assess the quality of mined passage mining finds a similar number of pseudo passage pairs, the labelling accuracy does increase from 28.0% to 47.0%. This is because the better pseudo parallel data improves the retriever, and the stronger retriever results in a better generator, which in turn leads to more relevant passages.
Figure 3: Performance of passage retrieval and keyphrase generation on EcommerceMKP, with different number of initial parallel data for iterative training.

Figure 4: Case study on the EcommerceMKP dataset. The present keyphrases (keyphrases shown in the description) are in bold while absent keyphrases are in italics. Correct predictions are in green while wrong predictions are in red.

7.3 Case Study

Fig. 4 exhibits an example of our model’s prediction. Given a product description in German, the model retrieves several English keyphrases and generates keyphrases in German accurately (translations in English are also provided). Through this example, we find that the retrieved English keyphrases do provide certain useful information such as “steiff teddy bear”, “grey” and “soft”, while it also brings some noise such as “my first”, “sweet-heart” and “vintage”. Although there is a wrong prediction “jimmy” caused by the retrieved English keyphrases, the improvement in results shows that the benefits of retrieved knowledge outweigh the noise it introduces. Moreover, the retrieved keyphrases are only regarded as a supplement to the original passage, and the generator can automatically focus on the informative parts from both inputs through self-attention. Our retrieval-augmented multilingual keyphrase generation can tolerate some noise from the retrieved English keyphrases and predict better keyphrases based on these external knowledge.

8 Conclusion

In this study, we investigate a novel task setting – multilingual keyphrase generation – and contribute two new multilingual keyphrase generation datasets covering multiple domains and languages. Furthermore, we propose a retrieval-augmented multilingual keyphrase generation framework with retriever-generator iterative training. Results show that the proposed approach outperforms a wide range of baselines.

References


Jingbo Shang, Jialiu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. IEEE Transactions on Knowledge and Data Engineering, 30(10):1825–1837.


A Appendix

A.1 Implementation Details

In the cross-lingual dense passage retriever, we use “bert-base-multilingual-cased” model (Wolf et al., 2020) to initialize the query and passage encoders and fine-tune it for 15 epochs with a batch size of 32. We share the parameters between the query encoder $E_Q(\cdot)$ and the passage encoder $E_P(\cdot)$ and map English and non-English passages into the same embedding space. Empirical results show the encoder with parameter sharing can perform slightly better. The positive examples are the corresponding English passages while we randomly sample 100 passages as negative examples in training. For the retrieval-augmented keyphrase generator, we fine-tune “mbart-large-cc25” (Wolf et al., 2020) for 10 epochs with Adam (Kingma and Ba, 2015) optimizer with a learning rate of 1e-4, a batch size of 8, a warm-up rate of 50 training steps. Similar to most Seq2Seq models, we train the mBART-based generation module by optimizing the negative log-likelihood loss of the ground-truth keyphrase sequence, and use beam search decoding with a beam size of 5 during inference. The number of retrieved keyphrases $m$ for retrieval-augmented generation is a hyperparameter and is tuned on the development set. We use keyphrases from $m = 1$ English passages for AcademicMKP dataset and $m = 5$ for EcommerceMKP dataset. During inference, we set the maximum target sequence length as 128 and set the beam decoding size as 5. For parallel passage mining via iterative training, we continue the iterative process until the retrieval recall does not improve. The total number of iterations ($T$ in Algo. 1) are 6 and 3 on EcommerceMKP and AcademicMKP respectively. The threshold $\tau$ in Line 23 for Algo. 1 is set as 5.

A.2 Variants of Retrieval Targets

There exists a misalignment between the retriever and the generator model. The retriever retrieves similar passages while the generator utilizes the associated keyphrases of these passages (not the retrieved passages) as external knowledge for generation. Therefore, a good retriever does not necessarily guarantee the good quality and usefulness of these keyphrases.

We tried two different retrieval targets which might close the misalignment. Given a non-English passage, we tried to either directly retrieve English keyphrases (RAMKG-P2K) or retrieve the concatenated sequences of passage-keyphrase pair (RAMKG-P2PK). We find that (1) RAMKG-P2K that directly retrieves keyphrases has poor retrieval performance. This is because it is hard to capture the similarity between non-English passages and English keyphrases; (2) RAMKG-P2PK has slightly worse results than only retrieval EN passages, which implies that additionally adding keyphrases in the retrieval targets does not bring any benefit.

Results are shown in Table 6. RAMKG (P2P) is our original model which retrieves English passages given a non-english passage. Results tell us that 1) directly retrieval of keyphrases have poor retrieval performance. This is because it is hard to capture the similarity between non-english passages and english keyphrases; 2) RAMKG (P2PK) has slightly worse results than the model , which implies that additionally adding keyphrases in the retrieval targets does not bring any benefit.

A.3 Discussion on Retriever-Generator Iterative training (RGIT) Algorithm

Difference between RGIT and Self-Training. Our approach shares some similarities with self-training (Lee, 2013; Pham et al., 2021) but there are some differences. In self-training, the teacher and student models are in the same architecture and focus on the same training objectives. In our proposed retriever-generator iterative training, the retriever and generator are two different models and optimized by different objectives.

Threshold Tuning. In this section, we investigate the impact of the chosen threshold $\tau$ (line 24) in our proposed retriever-generator iterative training. We tune the threshold (tau) on AcademicMKP and results are shown in Table 7. Results show...
that $\tau=5$ receives the best retrieval performance. $\tau=0$ brings more pseudo parallel passage pairs but introduces more noise. Larger $\tau$ (10/15) reduces the number of pseudo pairs, making the iterative training less effective.