AdaptKeyBERT: An Attention-Based approach towards Few-Shot & Zero-Shot Domain Adaptation of KeyBERT

Aman Priyanshu[1*], **Supriti Vijay**[2*],

[1]Department of Information & Communication Technology
[2]Department of Computer Science
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Karnataka, India.
supriti.vijay@gmail.com

Abstract

Keyword extraction has been an important topic for modern natural language processing. With its applications ranging from ontology generation, fact verification in summarized text, and recommendation systems. While it has had significant data-intensive applications, it is often hampered when the data set is small. Downstream training for keyword extractors is a lengthy process and requires a significant amount of data. Recently, Few-shot Learning (FSL) and Zero-Shot Learning (ZSL) have been proposed to tackle this problem. Therefore, we propose AdaptKeyBERT, a pipeline for training keyword extractors with LLM bases by incorporating the concept of regularized attention into a pre-training phase for downstream domain adaptation. As we believe our work has implications to be utilized in the pipeline of FSL/ZSL and keyword extraction, we open-source our code as well as provide the fine-tuning library of the same name AdaptKeyBERT at https://github.com/AmanPriyanshu/AdaptKeyBERT.

Introduction

The process of choosing the most important, pertinent, and descriptive terms from a single text to use as keywords is known as keyword extraction. In order to effectively transmit the informative content load of a document, keywords must represent distinct and specialized notions. Distinct keywords reduce redundancy in overall keywords extracted, whereas specialization assures quality keywords. However, with its varied applications in the fields of Bio-Medicine, Legal Entity Extractions, and News Verification among others, keyword extractors have become an integral segment of automated knowledge extraction systems. They've seen implementations for a varied resource of tasks such as summarization, indexing, contextual advertising, and personalized recommendations. Modern techniques have employed large-scale language model architectures, most having transformers as their backbone. Despite being quite effective at completing said task, they do need a considerable amount of labelled data for training, which may be difficult to source in particular low-resource domains and languages. Few-Shot Learning (FSL) & Zero-Shot Learning (ZSL) offers an augmentable solution to the aforementioned problem.
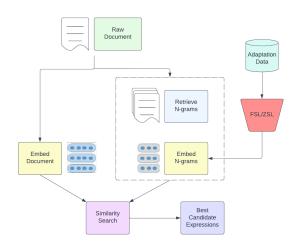
---

[*]These authors contributed equally.

Figure 1: Overall framework for zero-shot or few-shot integration to KeyBERT.

.

To address this, we propose AdaptKeyBERT, which aims to integrate few-shot andd zero-shot learning through attention mapping over candidate embeddings. We present the overall progression of few-shot and zero-shot domain adaptation in Figure 1. By incorporating regularized attention within KeyBERT's training module, we streamline the process of domain adaptation. We provide two variations to the algorithm, allowing users to encompass both (1) Few-Shot Domain Adaptation (2) Zero-Shot Domain Inclination.

An empirical study is then conducted to verify the nature of knowledge adapted after domain adaptation. Our objectives of achieving improved performance for both few-shot and zero-shot adaptation are formulated and described in detail below. Unlike existing frameworks for keyword extraction, AdaptKeyBERT exploits the significantly high performance of pre-trained LLM models present within KeyBERT, while improving them to qualitatively retrieve domain-specific keywords. Our contributions to this manuscript can be summarized as:

1. We reconsider downstream training keyword extractors

on varied domains by integrating pre-trained LLMs with Few-Shot and Zero-Shot paradigms for domain accommodation. We incorporate regularized attention based embedding reconstruction for domain attentive keyword extraction.

2. Demonstrate two experimental setting with the objectives of achieving high performance for (a) Few-Shot Domain Adaptation, & (b) Zero-Shot Domain Adaptation. The experimental results on both objective benchmarks demonstrate that our framework outperforms the base/naive approaches.

3. We open source a python library (AdaptKeyBERT) for the construction of FSL/ZSL for keyword extraction models that employ LLMs directly integrated with the KeyBERT API. Allowing easy training, validation, and deployment of said models.

Related Work

Keyword Extraction

Keyword extraction has become an indispensable tool for natural language processing. It has vast applications in downstream tasks like information retrieval systems - automatic text detection in historical manuscripts, slot filling, and extracting relations between entities, to name a few.

Keyword extraction approaches vary from supervised to unsupervised methods, wherein the supervised techniques require a substantial amount of labelled data which proves to be expensive. On the other hand, unsupervised approaches are utilized in a low-resource scenario. Past work has focussed chiefly on unsupervised techniques, employing statistical (KPMiner (El-Beltagy and Rafea 2009), RAKE (Rose et al. 2010), and YAKE (Campos et al. 2020)), graph-based (TextRank (Mihalcea and Tarau 2004)), embeddings-based (Key2Vec (Mahata et al. 2018), and EmbedRank (Rose et al. 2010)), KeyBERT (Grootendorst 2020)) and language-based methods (Tomokiyo and Hurst 2003). While supervised methods mainly delve into transformer architectures (Vaswani et al. 2017).

Out of all the state-of-the-art methodologies, KeyBERT(Grootendorst 2020) has been widely accepted for its better performance. It leverages pretrained BERT-based embeddings for keyword extraction. These are ranked according to the cosine similarity and compared to the embedding of the entire document. Due to the better performance, we employ KeyBERT in our pipeline for keyword extraction.

Keyword extraction with Zero-Shot Learning and Few Shot Learning

Recent works have incorporated keyword extraction with zero-shot learning for document filtering, text detection and crosslinguistic(BiKEA (Huang et al. 2014), Takasu (Takasu 2010)). (Tigunova et al. 2020) uses CHARM, a neural approach to identify salient keywords for a specific attribute. While (Song et al. 2020) proposes AGMHT, a zero-shot multi-label classifier that generates semantically meaningful features for zero-shot codes and reconstructs the code-relevant keywords. (Koloski et al. 2022) employ zero-shot cross-lingual keyword detection and compare performance

statistics amongst several state-of-the-art unsupervised keyword extractors.

Keyword extraction with few-shot learning has been marginally explored. (Gong and Eldardiry 2021) detects new relations from unlabelled data, which utilizes weighted side information constructed from labels, keywords and hypernyms of entities. (Liu et al. 2020) introduces an attention-based deep relevance model for few-shot document filtering, ADRM. This outperforms DAZER (Li et al. 2018), a classifier with the same architecture but employed with a zero-shot setting.

In our pipeline, we employ both few-shot and zero-shot settings on our domain attentive keyword extractor to address the lack of labelled data in an unsupervised approach. To our knowledge, this is the first methodology that compares both settings (FSL & ZSL) and incorporates them directly into keyword extraction architecture. Past work has only explored the integration of zero-shot learning in keyword extractors for cross-lingual datasets (Koloski et al. 2022).

Methodology

Problem Formulation

As illustrated in Figure 1, AdaptKeyBERT encapsulates zero-shot and few-shot learning by recomputing the embedding vectors of the candidate words. We first detail the notations and functioning of the baseline KeyBERT algorithm. For a keyword extraction task, we consider a target document $\mathbf{D}$ from which we aim to extract relevant keywords. $\mathbf{D}$ consists of multiple words/phrases, each of which could be potentially relevant keyword. Therefore, KeyBERT preliminarily employs n-gram based feature extraction. Here, we define the range of the n-gram extractor as, $[n_{ini}, n_{fin}]$. It defines the lower and upper boundary of the range of n-values for different word n-grams or char n-grams to be extracted. Once extracted we consider these as samples for the candidate set, $\mathbf{C} = \{c_1, c_2, c_3, ..., c_k\}$. Therefore, the task now encapsulates into ranking the above candidates in order of relevancy, such that they're still distinct from one-another. To solve the *ranking* function KeyBERT employs a Sentence Embedding network, **EModel**, which encodes sentences/phrases into high dimensional vectors. This in-turn is used to compute the latent vector representation of the document, $\mathbf{D} \rightarrow E_D$, and the candidate set, $\mathbf{C} \rightarrow E_C$. Where, $E_C = \{e_{c_1}, e_{c_2}, e_{c_3}, ..., e_{c_k}\}$ computed for each candidate phrase present. We then aim to maximize the ability of the ranking system based on their similarity to the document embedding.

Few-Shot Integration Given a keyword extraction task, it becomes integral to consider domain adaptation as an integral aspect of the field. With the generalization of Sentence Embedding Networks over all fields, it also reflects poorly on disproportionately marginalized topics. For example, we do not expect certain topics, such as core biology taxonomy, to be present in normal training sets or words which are relevant to legal artefacts. In such cases, the ability to rank certain domain words higher would be highly beneficial. There-

fore, a methodology to encompass the above objective is required, with same a smaller sample space.

Given the above KeyBERT module with an objective of few-shot domain adaptation, we employ attention to re-compute candidate-set embeddings. We first create a pre-training step, where the user must feed the model with sample $(\mathbf{D}, \mathbf{RC})$ tuples. Where each document, $\mathbf{D}$ is paired with its own list of relevant candidate phrases, $\mathbf{RC} \rightarrow \{rc_1, rc_2, rc_3, ...rc_l\}$. The embedding extraction mechanism of KeyBERT remains the same during this phase, however, we append a trainable *attention-layer* to the model, which aims to optimize the following function:

$$L(\theta)_{c \in RC} = \sum_{i=1}^{l} MSE(E_D, a_{c_i}) \qquad (1)$$

Where the model learns to adapt the attention-computed embedding vector (a_c) of relevant keywords to be closer to the document embeddings, $\mathbf{D}$, which in-turn are domain relevant. This allows the model to adapt to the topical shit. At the same time, we do not wish dissociate the embeddings of the other keywords, as they may be relevant in their own right. At the same time, restricting them to their original latent representation, allows AdaptKeyBERT to ensure that the attention doesn't simply move all the computed vectors towards the document embedding without semantic context. For this we employ the loss,

$$L(\theta)_{c \notin RC} = \sum_{i=1}^{l} MSE(a_{c_i}, e_{c_i}) \qquad (2)$$

This allows the attention mechanism to re-compute the embedding vector such that relevant keywords E_{RC} have an latent representation closer to the document embedding E_D, thus giving them an advantage over other candidate sets, which remain the same.

Zero-Shot Integration With the construction of a few-shot methodology for domain-adaptation, we observe that certain words can simply account as seed for words for particular domains, $\mathbf{SC} \rightarrow \{sc_1, sc_2, sc_3, ..., sc_m\}$. However, the methodology currently presented in KeyBERT among other keyword extraction techniques employing LLMs employ weighted mean computation between the original document embedding and the seed word embeddings (E_{SC}). Therefore, AdaptKeyBERT employs attention-computed weights to ascertain the relevance of candidate words over the seed words. A maximal score is then taken and used as the *similarity-weight*, sw, for re-computing the embeddings, a_c. A regularizer, $\alpha \in [0, 1]$, for reconfiguring the significance of the zero-shot integration methodology is also defined. We define this by,

$$a_c = (1 - sw * \alpha) * e_c + (sw * \alpha) * E_D \qquad (3)$$

Implementation Details

We build on both the objectives mentioned above, making sure AdaptKeyBERT delivers for both few-shot learning as well as zero-shot learning. For a given task, we ascertain the

most popular words presented in the dataset. These are directly fed into the zero-shot module as seed words for its integration. On the other hand, we retrieve those samples which encompass most of the these popular keywords within themselves from the dataset for few-shot learning. A popular keyword here is defined as being present in more than $p\%$ of the dataset (here, $p\% = 10\%$). For few-shot learning we restrict number of samples fed for domain adaptation to $p\%$ as well. Our experiments demonstrate the ability of the model to generalize on new toics only from a subset of the original data.

Experiments

Datasets and Experimental Settings

We categorize our experiments into two benchmarks based on the aforementioned objectives. We discuss the **fao780 dataset** (Food and Agriculture Organization). Which consists of 780 documents from the food and agriculture industry. We also take a look at the **CERN-290 dataset** which is composed of 290 high energy physics documents. We first quantify the most popular words in each dataset, allowing is to leverage AdaptKeyBERT to better understand domain adaptation for these particular domains.

Results

Model	Precision	Recall	F-Score
Benchmark	36.74	33.67	35.138
Zero-Shot	37.25	38.59	37.908
Few-Shot	40.03	39.1	39.559
Zero-Shot & Few-Shot	40.02	39.86	39.938

Table 1: AdaptKeyBERT performance on FAO-780 dataset with $p\% = 10\%$.

We perform a progressive learning routine to compute the performance of AdaptKeyBERT against the preliminary approach of KeyBERT. For the formulation of naive approach, we assume SentenceEmbeddings from the $all-MiniLM-L6-v2$ as the baseline model. We benchmark the performance of this model for comparison. We present results over both the datasets, FAO-780 dataset and the CERN-290 dataset.

We realize congruent results for both the datasets. Focusing on the FAO-780 results presented in Table 1 we can see a distinguishing improvement. While the Zero-Shot model weakly outperforms its benchmark counterpart compared to the Few-Shot approach. There's an improvement of 7.88% and 12.58% F1-Score respectively. On the contrary however, the combination of the two methodologies fails to improve drastically over the performance of the Few-Shot module. This may have been caused due to unintentional bias that similarity computation might have enabled contributed by the Zero-Shot framework. Even so, the amalgamation of both techniques achieves an improvement of over 13.66%.

Taking into consideration the CERN-290 dataset, we discover drastic improvements over the precision metric. How-

Model	Precision	Recall	F-Score
Benchmark	24.74	26.58	25.627
Zero-Shot	27.35	25.9	26.605
Few-Shot	29	27.4	28.177
Zero-Shot & Few-Shot	29.11	28.67	28.883

Table 2: AdaptKeyBERT performance on CERN-290 dataset with $p\% = 10\%$.

ever, the same fails for the recall as the number of False-Negatives increases in this benchmark. Even so, we achieve considerable performance for the same, improving F1-Score by 3.81%, 9.95%, and 12.7% respectively for Zero-Shot, Few-Shot, Zero+Few-Shot integration.

Conclusion

With an objective to adapt over new & disproportionately represented domains, we introduce AdaptKeyBERT, a first of its kind open-sourced pipeline which extends the Key-BERT library. By providing results across two liguistic-heavy datasets focused on different domains, we validate the performance and adaptation ability of our proposed methodology. Since keyword extraction models are often the first step in the construction of knowledge bases as well as the identification of certain information, we believe AdaptKey-BERT will be able to save a significant amount of memory and time computationally. We acknowledge that our keyword extraction pipeline may be susceptible to traditional biases plaguing similar ideations (Firoozeh et al. 2020), which we aim to address in future work.

References

Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; and Jatowt, A. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509: 257–289.

El-Beltagy, S. R.; and Rafea, A. 2009. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems*, 34(1): 132–144.

Firoozeh, N.; Nazarenko, A.; Alizon, F.; and Daille, B. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3): 259–291.

Gong, J.; and Eldardiry, H. 2021. Zero-shot Relation Classification from Side Information. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM.

Grootendorst, M. 2020. KeyBERT: Minimal keyword extraction with BERT.

Huang, C.-C.; Eskenazi, M.; Carbonell, J.; Ku, L.-W.; and Yang, P.-C. 2014. Cross-Lingual Information to the Rescue in Keyword Extraction. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 1–6. Baltimore, Maryland: Association for Computational Linguistics.

Koloski, B.; Pollak, S.; Škrlj, B.; and Martinc, M. 2022. Out of Thin Air: Is Zero-Shot Cross-Lingual Keyword Detection Better Than Unsupervised?

Li, C.; Zhou, W.; Ji, F.; Duan, Y.; and Chen, H. 2018. A Deep Relevance Model for Zero-Shot Document Filtering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2300–2310. Melbourne, Australia: Association for Computational Linguistics.

Liu, B.; Li, C.; Zhou, W.; Ji, F.; Duan, Y.; and Chen, H. 2020. An Attention-Based Deep Relevance Model for Few-Shot Document Filtering. *ACM Trans. Inf. Syst.*, 39(1).

Mahata, D.; Kuriakose, J.; Shah, R. R.; and Zimmermann, R. 2018. Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 634–639. New Orleans, Louisiana: Association for Computational Linguistics.

Mihalcea, R.; and Tarau, P. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. Barcelona, Spain: Association for Computational Linguistics.

Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. *Automatic Keyword Extraction from Individual Documents*, chapter 1, 1–20. John Wiley & Sons, Ltd. ISBN 9780470689646.

Song, C.; Zhang, S.; Sadoughi, N.; Xie, P.; and Xing, E. 2020. Generalized Zero-Shot Text Classification for ICD Coding. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 4018–4024. International Joint Conferences on Artificial Intelligence Organization. Main track.

Takasu, A. 2010. Cross-Lingual Keyword Recommendation Using Latent Topics. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, 52–56. New York, NY, USA: Association for Computing Machinery. ISBN 9781450304078.

Tigunova, A.; Yates, A.; Mirza, P.; and Weikum, G. 2020. CHARM: Inferring Personal Attributes from Conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5391–5404. Online: Association for Computational Linguistics.

Tomokiyo, T.; and Hurst, M. 2003. A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, 33–40. Sapporo, Japan: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need.