# Multi-task Learning Based Keywords Weighted Siamese Model for Semantic Retrieval

Mengmeng Kuang[1], Zhenhong Chen[1], Weiyan Wang[1,2](✉), Lie Kang[1],
Qiang Yan[1], Min Tang[1], and Penghui Hao[1]

[1] Tencent Holdings Ltd., Shenzhen, China
{mengkuang,hollischen,liekang,rolanyan,kevinmtang,terryhao}@tencent.com
[2] Hong Kong University of Science and Technology, Hong Kong, China
wwangbc@cse.ust.hk

**Abstract.** Embedding-based retrieval has drawn massive attention in online search engines because of its semantic solid feature expression ability. Deep Siamese models leverage the powerful dense embeddings from strong language models like BERT to better represent sentences (queries and documents). However, deep Siamese models can suffer from a sub-optimal relevance prediction since they can hardly identify keywords due to late interaction between the query and document. Although some studies tried to adjust weights in semantic vectors by inserting some global pre-computed prior knowledge, like TF-IDF or BM25 scores, they neglected the influence of contextual information on keywords in sentences. To retrieve better-matched documents, it is necessary to identify the keywords in queries and documents accurately. To achieve this goal, we introduce a keyword identification model to detect the keywords from queries and documents automatically. Furthermore, we propose a novel multi-task framework that jointly trains both the deep Siamese model and the keywords identification model to help improve each other's performance. We also conduct comprehensive experiments on both online A/B tests and two famous offline benchmarks to demonstrate the significant advantages of our method over other competitive baselines.

**Keywords:** Text matching · multi-task learning · Siamese model · semantic retrieval · keywords identification

## 1 Introduction

In the era of information explosion, it is more and more critical to quickly and accurately find query-related information from a large number of documents. Representation learning based retrieval has impressively improved the retrieval accuracy and reformed this critical field researched for decades [25]. Based on the deep matching models [4] and the state-of-the-art pre-trained frameworks, semantic retrieval has thrived as a typical application of representation learning.

H. Kashima et al. (Eds.): PAKDD 2023, LNAI 13937, pp. 86–98, 2023.
https://doi.org/10.1007/978-3-031-33380-4_7

An extensive collection of works, especially the deep Siamese models [4,6], have been proposed to tackle the semantic retrieval task [11]. DSSM [8], CLSM [23], ARC-I [5] explore adopting the traditional neural networks, while Sentence-BERT [18], ColBERT [10], TwinBERT [13] take a further step to employ the pre-trained language models like BERT [2]. All these works highlight the charm of deep Siamese structures. Especially, pre-trained BERT can effectively capture the contextual semantic meanings in the query or document with the self-attention [9], which significantly enhances the accuracy of the Siamese semantic retrieval model.

However, it is hard for deep Siamese models to directly infer the keywords in the query since there is no interaction between the query and the document. Keywords have been proven to play a unique and important role in information retrieval applications [19]. To realize the pre-computation for massive documents, the deep Siamese structure has the independent query encoder and document encoder, which have no interaction until the last layer computing the similarity. But unfortunately, the query encoder itself can have difficulties in adequately weighting different words in the query without any context information about documents. And the document encoder has a similar problem without any query information. Therefore, semantic representations without keyword identification will directly impact semantic similarity computing and thus affect the overall matching process.

In many previous studies, the global statics of context information is used to improve the query representations by introducing some pre-computed prior knowledge, like BM25 [19] or TF-IDF [20] scores. However, such statics can not take the contextual semantics into consideration to reflect the word weights precisely. A word is significant in one sentence, but may be not in another. Apparently, if we use the pre-computed statics as the prior knowledge, it can lead to a sub-optimal and even poor decision.

To remedy the limitations, we introduce a multi-task learning based keywords weighted Siamese model (MKSM) for semantic retrieval in this work. We propose a novel keywords identification model joined with the Siamese retrieval model to explicitly model the weights of the adaptable keywords and get better representations for the retrieval. Specifically, we model the keyword identification as a regression learning problem to consider contextual semantics instead of rule-based statistics. Furthermore, The keyword identification model shares the same neural network model with the Siamese model but has different training loss functions. Therefore, we train both the keyword identification model and the deep Siamese model jointly in the style of multi-task learning to improve each other's performance. The multi-task learning enables our solution to learn better keyword weights from retrieval signals and the regression target. Therefore, we can get a better representation containing the semantic meaning of keyword weights to conduct the matching process better.

To verify its effectiveness, we evaluate our proposed MKSM in the online production environment and on famous and public benchmark datasets. Specifically, MKSM has been deployed for the online service search scenario of a popular social application frequently used by over 100 million users. The online A/B test in real production shows that MKSM concretely improves the user experiences for the service search in terms of click-through rate (CTR) and retrieval rate (RR) for real production. Furthermore, the empirical results on public searching benchmarks have also demonstrated considerable improvements over baselines.

To summarize, our contributions are three-fold:

- We introduce an adaptable keywords identification model to learn better representations for queries and documents.
- We propose a novel semantic retrieval framework MKSM which joins the keywords identification method to a Siamese model for semantic retrieval in the form of multi-task learning.
- Extensive experiments on online A/B tests and two offline public benchmarks verify the effectiveness of our proposed model.

## 2 Related Works

A variety of deep matching models have been proposed for the information retrieval problems [15]. Siamese models applied to semantic retrieval started from the Deep Structured Semantic Model (DSSM) [8], which mapped both query and document to the same semantic space, and achieved the purpose of retrieval by maximizing the cosine similarity. Then, ARC-I [5] and CLSM [23] used Convolutional neural networks (CNNs) and max pooling to replace the fully connected networks to extract features, which could capture more contextual information for semantic vector representation. Further, LSTM-DSSM [17] proposed to use Long Short-term Memory (LSTM) networks to replace CNNs to obtain contextual information over longer sequences accurately. Sentence-BERT [18], a modification of the pre-trained BERT network that used Siamese and triplet network structures to derive semantically meaningful sentence embeddings that could be compared using cosine-similarity, which reduced the effort of finding the most similar pair from 65 h with BERT/RoBERTa to about 5 s, while maintaining the accuracy from BERT. Recently, TwinBERT [13] used twin-structured BERT-like encoders to encode the query and document, respectively, and a crossing layer to combine the two embeddings to produce a similarity score. Additionally, ColBERT [10] introduced a late interaction architecture that independently encoded the query and the document using BERT and then employed a cheap yet powerful interaction step that modeled their fine-grained similarity. Furthermore, [14] proposed a simple neural model that combined the efficiency of dual encoders with some of the expressiveness of more costly attentional architectures and explored sparse dense hybrids to capitalize on the precision of sparse retrieval. Accurately representing the text and its contextual information has always been a hot research direction, which is our concern in MKSM. Neither the models mentioned above nor the [7] (Facebook), MOBIUS [3] (Baidu), etc. that have been applied in the actual business has learned global contextual word weights.

# 3 Methods

In this section, we first provide the problem definition of the semantic retrieval task. Then we propose a comprehensive overview of our framework and further introduce the implementations of each component in the MKSM framework. Finally, we describe the multi-task training process of our structure.

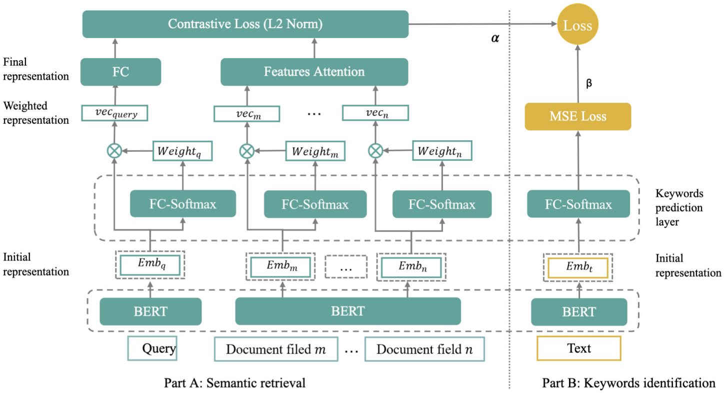


**Fig. 1.** The hierarchical framework of MKSM. *The whole framework can be divided into two parts. the left part, Part A, illustrates the semantic retrieval process, while the right part, part B, represents the keywords identification task.*

## 3.1 Problem Definition

The semantic retrieval task can be described as a matching problem $M$ that gives a matching score for each query $q$ and document $d$ pair. Here, we use a single symbol $d$ to stand the entire document which usually contains not only one field (e.g., name, description, etc.). Before calculating the matching score $m_s$, every string needs to be embedded as a semantic vector by some embedding methods $E$, like the BERT language model. In our framework, in addition to simply embedding the query and document, the keywords identification model can be regarded as an independent function $K$. Hence, the keywords weighted Siamese model for semantic retrieval can be represented as Eq. (1).

$$m_s = match(q, d) = M(K(E_q(q)), K(E_d(d))). \tag{1}$$

## 3.2 Framework Overview

From a horizontal view, MKSM comprises three parts, BERT semantic representation, Keyword weight correction, and Matching score calculation, as shown in Fig. 1. The framework, divided into semantic retrieval and keywords identification parts, starts with the query, document, and text represented by a shared BERT-pertained language model to get the corresponding embedding ($Emb$). A shared fully connected ($FC$) layer and softmax are appended to learn the word weights ($Weight$) in the keywords identification module. Moreover, the $Weight$ is utilized for weighing the embeddings of the query and various fields in the document to get better feature vectors ($vec$). Then, the contrastive loss (with $L_2$ normalization) is performed to measure the relevance of the query and document representations in the semantic retrieval part. The mean squared error ($MSE$) loss is applied as the objective of the keywords identification module. The total loss is the sum of the contrastive loss and $MSE$ loss weighted by two hyperparameters, $\alpha$, and $\beta$, respectively.

### 3.3 The Keywords Identification Model

Unlike other statistical methods or keyword detection methods, we model the keywords identification task as a regression task that fits word importance sequence $s_k$ from input sequence $s$.

For the offline public benchmark, we take all positive documents as one click and negative documents as no click to estimate the clicking rate. As shown in Fig. 2, we consider the clicking rates $a_i$ of documents and all related second-order queries $s'$ to generate the keyword weights of the first-order query $s$. Specifically, we generate labels as follows:

1. We collect a large amount of high-relevance retrieval logs containing the query $s$ and remove all stop words in the logs.
2. For the first-order query $s$, we dig out all clicked documents $d$ and their corresponding click rates $a$.
3. For any clicked document $d_j$, we find all second-order queries that retrieve it. And we think the queries retrieving the same document have similar semantic meanings.
4. Then we count the word frequency $f_{w_i}^{d_j}$ in the second-order queries of the document $d_j$ and then normalized all word frequencies as $f_{w_i}^{d_j} = f_{w_i}^{d_j} / \sum_i f_{w_i}^{d_j}$.
5. Finally, we combine all normalized word frequencies of all documents by weight averaging to generate the keyword weight as $f_{w_i} = \sum_j a_j f_{w_i}^{d_j} / \sum_i f_{w_i}^{d_j}$

Similarly, we generate the document keyword label by assuming that the clicked documents for the same query have similar semantic meanings. Specifically, we dig out all queries that retrieve the document and all other documents related to those queries. We count word frequencies in all related documents and then normalize the frequency according to click rates and the sum of frequencies.

As presented in the right part of Fig. 1, the keywords identification model includes three components, (1) representation component, (2) weight layer ($L_k$), and (3) loss calculation. We use the embedding of the "[CLS]" token in the BERT sentence embeddings as the initial representation. Then, a fully connected layer is supplemented with a hidden size equal to the padding size used to learn the word weights. Finally, the $MSE$ loss function is computed for optimizing the parameters in the weight layer by Eq. (2).

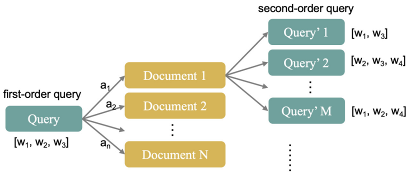


**Fig. 2.** Keyword weight label mining from second-order queries in history logs

$$loss_{keywords} = \frac{1}{P}\sum_{i=1}^{P}\left(l_i - L_k(BERT(s)_{[CLS]})_i\right)^2, \tag{2}$$

where $l$ stands for the observed values of $s$.

### 3.4 The Siamese Retrieval Model

Like most deep matching models [3,7,13,18,24], the retrieval model also employs a twin-structured Siamese framework as shown in the left part of Fig. 1. The structure is a two-part design formed by the representation part and the matching part. In the representation part, there are three layers of representation, (1) initial representation, (2) weighted representation ($L'_k$), and (3) final representation ($L_f$). The initial representation is the average BERT embeddings of all the tokens. The weighted representation ($L'_k$) is the initial representation associated with the weight layer ($L_k$) in the keywords identification model. The final representation is used as input of the matching part to calculate the matching scores minimized by a loss function, introduced in the rest of this section. We optimize the model to acquire a better matching score by minimizing the contrastive loss ($loss_{matching}$) as presented in Eq. (3).

$$loss_{matching} = \frac{1}{2N}\sum_{i=1}^{N} y(\mathrm{D}(r_q, r_d)_i)^2 + (1-y)\max(m - \mathrm{D}(r_q, r_d)_i, 0)^2, \tag{3}$$

where $y$ is the relevance label with equals to 1 (relevant) or 0 (irrelevant), D represents the Euclidean distance, which can be expressed by Eq. (4), and $m$ is a margin threshold.

$$\mathrm{D}(r_q, r_d) = \|r_q - r_d\|_2 = \left(\sum_{i=1}^{P}\left(r_q^i - r_d^i\right)^2\right)^{\frac{1}{2}} \tag{4}$$

### 3.5 The Multi-task Learning Strategy

As stated in Sects. 1, 3.3 and 3.4, to make the keywords identification model can learn adaptive keywords weights, we propose to train the keywords identification model and the Siamese retrieval model together. The combined loss function can be represented as Eq. (5).

$$loss = \alpha \times loss_{matching} + \beta \times loss_{keywords}. \tag{5}$$

where $\alpha$ and $\beta$ are two hyper-parameters.

The training repeats the following back-propagation processes until the Siamese model can learn the representations of queries and documents well.

1. Firstly, back-propagating on the keyword identification model.
2. Secondly, back-propagating on the Siamese retrieval model with fixed parameters of shared weight layer.

## 4 Experiments

In this section, we first introduce the details of the experiment settings. Then we discuss the experiment results, including offline performance, online evaluation, and ablation study. Case study results and discussion on our work are in the supplementary materials.

**Table 1.** Dataset Statistics

| Datasets | MS MARCO | Private |
|---|---|---|
| Training set | 367,000 queries<br>3,200,000 documents | 260,000 queries<br>1,300,000 documents |
| Validation set | 519,300 pairs | 80,000 pairs |
| Fields | title<br>body | account name<br>service name<br>service description |
| Average length | 1137 | 96 |

### 4.1 Experiments Setup

**Evaluation Datasets.** We validate the performance of MKSM on two datasets, where one is a public benchmark and the other is private. **MS MARCO** [16] is a famous benchmark from Microsoft, which is sampled from Bing's search query logs. To construct our **Private** dataset, we extract the daily service search logs from a popular instant messaging application and manually label the relevance. And we implement a label noise detection method based on confident learning [12] to purify this dataset. **MS MARCO** is an English dataset while **Private** is a Chinese benchmark. Table 1 summarizes the detailed statistics of such two datasets from three aspects.

**Evaluation Metrics.** Because our approach focuses on the matching stage in semantic retrieval, we choose Normalized Cumulative Gain (NCG) [22] as the evaluation metric. It is the best empirical metric for query-document matching, because it reflects the number of relevant documents returned without casing the specific ranking. NCG is computed as

$$\mathrm{NCG} = \frac{\mathrm{CG}}{i\mathrm{CG}}, \tag{6}$$

where CG(Cumulative Gain) is the sum of all the relevance scores in the recall set, and $i$CG is the ideal CG, which is the sum of relevance scores of the ideal document recall set. Specifically, CG is defined as

$$\mathrm{CG} = \sum_{i=1}^{T} relevance\_score_i, \tag{7}$$

**Baselines.** We compare our proposed MKSM framework with 6 representative retrieval baseline models[1]. Such methods can be categorized into different classes as follows (the detailed discussion of these methods is presented in Sect. 2),

- *Classical retrieval methods*: **TF-IDF** [20] and **BM25** [19].
- *Deep Siamese models*: **CLSM** [23] and **USE** [1].
- *Pre-trained language model*: **BERT** [2].
- *Keywords weighted model*: **BERT+TF-IDF** [21].

**Implementation Details** All the implementations mentioned in this paper are based on TensorFlow. We train MKSM with 4 NVIDIA Tesla V100 GPUs paralleled. The semantic vector representations for queries and documents are based on BERT pre-trained language model with padding size 128 or 1024 in the two datasets. The $\alpha$ and $\beta$ are set as 0.6 and 0.4, respectively. AdamW, as an improved Adam optimizer, is used in the training processes in the MKSM framework. To make the inference phase more efficient, the embedding of documents is offline performed ahead. The cosine similarity of the query and document representations is utilized as the matching score.

### 4.2 Overall Performance

Table 2 illustrates the comparison results of NCG@$T$, where underlined numbers are the best results of baselines and bold numbers are the best results of all models. The difference between MKSM and MKSM$_{\mathrm{SEP}}$ is whether training the keywords identification model and the Siamese retrieval model separately.

From the results of Table 2, we can conclude the following observations,

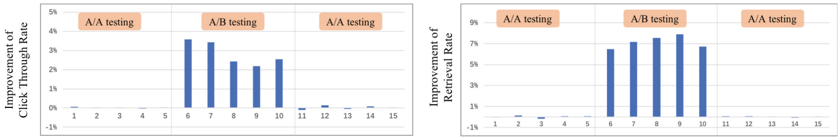
[1] We don't compare with other baselines listed in the Sect. 2 since they are not open-sourced or fine-tuned for different retrieval scenarios.

- Our proposed MKSM framework obtains the best performance over other retrieval models in both benchmarks. Specifically, in MS MARCO and Private datasets, MKSM receives at least 0.9% and 0.7% promotion in terms of NCG, respectively. These results indicate the superiority of our proposed MKSM framework over baselines in both English and Chinese benchmarks, with different lengths of documents.

**Table 2.** Overall performance on MS MARCO and Private datasets

| Models | MS MARCO | | | | Private | | | |
|---|---|---|---|---|---|---|---|---|
| | NCG@10 | NCG@20 | NCG@50 | NCG@100 | NCG@5 | NCG@10 | NCG@20 | NCG@30 |
| TF-IDF | 0.4154 | 0.5178 | 0.6258 | 0.7158 | 0.8398 | 0.8752 | 0.9190 | 0.9455 |
| BM25 | 0.4360 | 0.5465 | 0.6736 | 0.7564 | 0.8332 | 0.8674 | 0.9158 | 0.9431 |
| CLSM | 0.4016 | 0.5245 | 0.6541 | 0.7155 | 0.7446 | 0.8146 | 0.8930 | 0.9330 |
| USE | 0.3746 | 0.4045 | 0.6045 | 0.6620 | 0.8376 | 0.8784 | 0.9253 | 0.9521 |
| BERT | 0.4574 | 0.5745 | 0.6920 | 0.7841 | 0.8332 | 0.8763 | 0.9186 | 0.9487 |
| BERT+TF-IDF | 0.4562 | 0.5771 | 0.6938 | 0.7864 | 0.8432 | 0.8773 | 0.9268 | 0.9507 |
| $\text{MKSM}_{\text{SEP}}$ | 0.4619 | 0.5809 | 0.6992 | 0.7896 | 0.8452 | 0.8841 | 0.9302 | 0.9582 |
| MKSM | **0.4630** | **0.5868** | **0.7041** | **0.7934** | **0.8521** | **0.8904** | **0.9336** | **0.9621** |
| Impr | 1.2% | 1.7% | 1.5% | 0.9% | 1.1% | 1.5% | 0.7% | 1.1% |

"Impr." presents the improvement of MKSM over the best baseline.



(a) Online experimental results of click-through rate.

(b) Online experimental results of retrieval rate.

**Fig. 3.** Online evaluations.

- BERT+TF-IDF performs better than BERT in most empirical metrics, which indicates that the leverage of prior knowledge in query representations significantly improves the retrieval performance. Besides, our proposed MKSM and $\text{MKSM}_{\text{SEP}}$ both perform better than BERT+TF-IDF. It demonstrates that keyword identification performs better than the traditional statistical information TF-IDF as the prior knowledge, no matter whether in separating training or multi-task training. It is because that our proposed keywords identification model can provide the keywords weight information, which is essential in the retrieval task.
- MKSM performs better than $\text{MKSM}_{\text{SEP}}$ in terms of all metrics, which indicates that the training strategy of MKSM can influence the performance, and multi-task learning can introduce the prior knowledge to the Siamese model effectively.

### 4.3 Online Evaluation

We conduct A/B testing in the service retrieval scenario, comparing the proposed model MKSM with the current baseline, a distilled BERT.

The whole online experiment lasts 15 days. We monitor the results of A/A testing for the first five days, conduct A/B testing for the following five days, and conduct A/A testing again in the last five days. 15% of the users are randomly selected as the experimental group, and another 15% of the users are in the control group. During A/A testing, all the users are served by the BERT. During A/B testing, users in the control group are presented with retrieval results by the BERT, while users in the experimental group are presented with the MKSM semantic retrieval results. Note that the click experiment of MKSM shares the same exposure with the distilled BERT to verify whether the improvement is caused by the new semantic retrieval design.

Figures 3(a) and 3(b) show the improvement of the experimental group over the control group with respect to click-through rate (CTR) and retrieval rate (RR), which are defined as Eq. (8). We can see that the system is relatively stable in terms of CTR and RR during the A/A testing. As for the A/B testing, which starts from day 6, a significant improvement over the baseline BERT can be clearly observed. Specifically, the improvements concerning CTR and RR received by MKSM are at least 2% and 6%, respectively. In the final five days, we conduct A/A testing again, which replaces the MKSM framework with the distilled BERT. The improvement obtained by MKSM decays rapidly, which further proves the effectiveness of A/B testing.

$$\mathrm{CTR} = \frac{\#click}{\#exposure}, \mathrm{RR} = \frac{\Delta exposure}{\#exposure}, \tag{8}$$

where # means the number of *click* and *exposure*, and *Δexposure* means the increment of good results in *exposure*.

### 4.4 Ablation Study

In this subsection, to study the effectiveness of each component and certify that MKSM is the best combination, we conduct several models which are different from MKSM in terms of each component, such as $\mathrm{MKSM}_{\mathrm{SEP}}$, BERT, and $\mathrm{MKSM}_{[\mathrm{CLS}]}$. Specifically, $\mathrm{MKSM}_{\mathrm{SEP}}$ trains the keywords identification model and the Siamese model separately. BERT is the pure pre-trained language model without any prior knowledge. $\mathrm{MKSM}_{[\mathrm{CLS}]}$ uses the embedding of the [CLS] token in BERT as the initial representation of queries and documents. The performance comparison in **Private** benchmark is presented in Table 3.

From the results, we can confirm that:

- Compared with $\mathrm{MKSM}_{\mathrm{SEP}}$, MKSM shows a better performance, which indicates the multi-task learning manner can make the Siamese model and the keywords identification model interact more effectively.

- To demonstrate the superiority of the keywords identification model, we compare MKSM with BERT, a pure pre-trained language model for retrieval tasks without prior knowledge. The results reflect the effectiveness of our proposed keywords identification model. Besides, in contrast to fixed statistical information(referred to as "BERT+TF-IDF" in Table 2), our proposed MKSM with keywords identification model shows a better retrieval performance, which further indicates the superiority of our proposed MKSM framework.
- MKSM achieves better performance than $\text{MKSM}_{[\text{CLS}]}$. It means that using the average of all token embeddings as the initial representation of queries and documents in MKSM is slightly better than using only [CLS] token embedding. The reason is, compared with [CLS] token embedding, the average embedding of the BERT Encoder can provide much more useful information for keyword identification and retrieval.

**Table 3.** Ablation Study of MKSM

| Methods | NCG@5 | NCG@10 | NCG@20 | NCG@30 |
|---|---|---|---|---|
| $\text{MKSM}_{\text{SEP}}$ | 0.8452 | 0.8841 | 0.9302 | 0.9582 |
| BERT | 0.8332 | 0.8763 | 0.9186 | 0.9487 |
| $\text{MKSM}_{[\text{CLS}]}$ | 0.8447 | 0.8803 | 0.9277 | 0.9547 |
| MKSM | 0.8521 | 0.8904 | 0.9336 | 0.9621 |

## 5 Conclusion

In this paper, we propose a novel semantic retrieval model MKSM, which utilizes a keywords identification model and multi-task learning strategy to introduce practical prior knowledge in a Siamese model. MKSM can automatically learn the keywords in queries and documents by integrating the keyword weight layer and providing better final representations for calculating matching scores. We conduct extensive experiments and rigorous analysis in online A/B tests and offline public benchmarks to demonstrate that MKSM outperforms other modern deep matching models on semantic retrieval.

## References

1. Cer, D., et al.: Universal sentence encoder. arXiv (2018)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv (2018)
3. Fan, M., Guo, J., Zhu, S., Miao, S., Sun, M., Li, P.: Mobius: towards the next generation of query-ad matching in baidu's sponsored search. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2509–2517 (2019)

4. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for adhoc retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 55–64 (2016)
5. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural Information Processing Systems, pp. 2042–2050 (2014)
6. Huang, C., Liu, Q., Chen, Y.Y., et al.: Local feature descriptor learning with adaptive siamese network. arXiv (2017)
7. Huang, J.T., et al.: Embedding-based retrieval in facebook search. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2553–2561 (2020)
8. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 2333–2338 (2013)
9. Jawahar, G., Sagot, B., Seddah, D.: What does bert learn about the structure of language? In: ACL 2019–57th Annual Meeting of the Association for Computational Linguistics (2019)
10. Khattab, O., Zaharia, M.: Colbert: efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–48 (2020)
11. Klyuev, V., Oleshchuk, V.: Semantic retrieval: an approach to representing, searching and summarising text documents. Int. J. Inf. Technol. Commun. Convergence **1**(2), 221–234 (2011)
12. Kuang, M., Wang, W., Chen, Z., Kang, L., Yan, Q.: Efficient two-stage label noise reduction for retrieval-based tasks. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 526–534 (2022)
13. Lu, W., Jiao, J., Zhang, R.: Twinbert: distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM 2020, pp. 2645–2652. Association for Computing Machinery, New York (2020)
14. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Trans. Assoc. Comput. Linguist. **9**, 329–345 (2021)
15. Mitra, B., Craswell, N., et al.: An introduction to neural information retrieval. Now Foundations and Trends (2018)
16. Nguyen, T., et al.: Ms marco: a human generated machine reading comprehension dataset. In: CoCo@ NIPS (2016)
17. Palangi, H., et al.: Semantic modelling with long-short-term memory for information retrieval. arXiv (2014)
18. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. arXiv (2019)
19. Robertson, S., Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Now Publishers Inc., Norwell (2009)
20. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
21. Shan, X., et al.: Glow: global weighted self-attention network for web search. arXiv (2020)
22. Shan, X., et al.: Glow : global weighted self-attention network for web search. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 519–528 (2021)

23. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 101–110 (2014)
24. Sun, X., Tang, H., Zhang, F., Cui, Y., Jin, B., Wang, Z.: Table: a task-adaptive bert-based listwise ranking model for document retrieval. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2233–2236 (2020)
25. Xiong, L., et al.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv (2020)