Adaptive Feature Discrimination and Denoising for Asymmetric Text Matching

Yan Li [1], **Chenliang Li**[2], **Junjun Guo**[1, *]
[1] Kunming University of Science and Technology
[2] Wuhan University
{liyayli,guojjgb}@163.com
cllee@whu.edu.cn

Abstract

Asymmetric text matching has becoming increasingly indispensable for many downstream tasks (*e.g.,* IR and NLP). Here, *asymmetry* means that the documents involved for matching hold different amounts of information, *e.g.,* a short query against a relatively longer document. The existing solutions mainly focus on modeling the feature interactions between asymmetric texts, but rarely go one step further to recognize discriminative features and perform feature denoising to enhance relevance learning. In this paper, we propose a novel **a**daptive feature **d**iscrimination and **d**enoising model for **a**symmetric te**x**t matching, called **ADDAX**. For each asymmetric text pair, ADDAX is devised to explicitly distinguish discriminative features and filter out irrelevant features in a context-aware fashion. Concretely, a matching-adapted gating siamese cell (MAGS) is firstly devised to identify discriminative features and produce the corresponding hybrid representations for a text pair. Afterwards, we introduce a locality-constrained hashing denoiser to perform feature-level denoising by learning a discriminative low-dimensional binary codes for redundantly longer text. Extensive experiments on four real-world datasets from different downstream tasks demonstrate that the proposed ADDAX obtains substantial performance gain over 36 up-to-date state-of-the-art alternatives.

1 Introduction

Given a pair of documents, text matching aims to precisely predict the semantic relations between them. An efficient and effective matching algorithm is now an indispensable asset in many information retrieval, question answering and dialogue systems. In these application scenarios, a text pair in matching (*e.g.,* query-document and question-answer pair) ususlly has a large disparity

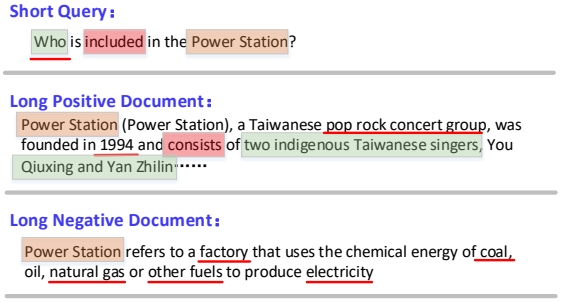


Figure 1: An example of asymmetric text matching.

in the quantity of information, *a.k.a.* asymmetric text matching. For example, a matching pair have 7.15 and 95.54 words respectively on average in InsuranceQA dataset (Feng et al., 2015) (*i.e.,* the difference being about an order of magnitude). This asymmetry between a short query and a long document renders it as a nontrivial task.

The existing solutions can be grouped into two categories, namely representation-based and interaction-based models (Khattab and Zaharia, 2020). The former category mainly utilizes convolutional neural networks (CNN) and recurrent neural networks (RNN) to learn the latent representation of a document independently, including DSSM (Huang et al., 2013), SNRM (Zamani et al., 2018). On contrast, the latter category focuses on leveraging fine-grained interaction signals between them. It is widely recognized that exploiting interaction signals would largely improve the relevance learning capacity. Examples include DRMM (Guo et al., 2016), KNRM (Xiong et al., 2017). Recently, with the prominence of deep pre-trained language models (LMs) like BERT (Devlin et al., 2019), uptodate LMs-based deep relevance models significantly push the frontier of the state-of-the-art further (Dai and Callan, 2019b; Nogueira and Cho, 2019; Xu and Li, 2020). Though significant performance gain is achieved by these efforts, they mainly overlook further feature discrimination and

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 1146–1156
October 12–17, 2022.

denoising between asymmetric texts, which can be potentially useful to enhance matching performance.

To explain this point, we give an illustrative example in Figure 1. Here, polysemous word like "*power station*" in the query side hinders the precise matching. On the other hand, the semantic association among "*who*", "*pop rock concert group*" and "*singers*" assists the relevance learning process. Also, word pair like "*who*" and "*factory*" describe two distinct things, which can be reflected via the feature-level interactions. Hence, recognizing the discriminative features and filtering out noisy features certainly enhance the relevance learning process.

To this end, in this paper, we propose an adaptive feature discrimination and denoising model for asymmetric text matching, named ADDAX. Specifically, ADDAX consists of a BERT-based context encoder, a matching-adapted gating siamese cell (called MAGS), a locality-constrained hashing denoiser, and a MaxSim (Khattab and Zaharia, 2020) based relevance predictor. For each document, we firstly derive the word-level contextual representations through a BERT-based context encoder. Afterwards, MAGS utilizes an cross-attention mechanism to represent a document with relevant information from its counterpart in the matching pair, which produces the word-level reference representations for the former. Then, the resultant word-level attention information is leveraged to discriminate the importance of these reference representations and their divergence against the orginal representations. We then utilize a highway network to adaptively composite these two kinds of semantic signals as the context-aware hybrid representations.

After this feature-level discrimination, we utilize a locality-constrained hashing denoiser to project the long document into low-dimensional binary space. The hashing denoiser is formulated as an autoencoder over the hybrid representations. That is, the semantics relevant to the text pair will be preserved by the denoising process, which further facilitates the relevance learning. Finally, a MaxSim operator (Khattab and Zaharia, 2020) is employed to calculate final relevance score. Overall, the key contributions are summarized as below:

- We propose an adaptive feature discrimination and denoising model for asymmetric text matching. To the best of our knowledge, ADDAX is the first attempt to explicitly derive discriminative features and perform feature denoising for this task.
- To derive discriminative features, a matching-adapted gating siamese cell (called MAGS) is devised to synthesize hybrid representations for a text pair in terms of word-level relevance information. To perform feature denoising, a locality-constrained hashing denoiser is devised to purify context-aware semantics and filter out feature-level noise for the long document.
- Extensive experiments are conducted on four real-world datasets and the results demonstrate the superior performance of our method compared against existing SOTA alternatives.

2 RELATED WORK

2.1 Traditional Neural Matching Models

Recent years, deep learning has delivered significant performance gain for various text matching tasks. Generally, deep relevance matching models can be divided into two categories. The first is the representation-based models that independently encodes a query and a document into two vectors and estimates relevance in terms of vector similarity (*e.g.,* DSSM (Huang et al., 2013), CDSSM (Shen et al., 2014), LSTM-RNN (Palangi et al., 2016)). For instance, Huang et al. (2013) propose to embed query and document into two vectors through a multilayer perceptron and calculates the corresponding cosine similarity as the matching score. Following this work, SNRM (Zamani et al., 2018) exploits the sparsity property to derive a sparse-vector representation for each query/document, which allows it to also leverage a dense index to do fast end-to-end retrieval. The second category, interaction-based models, exploits complex fine-grained interactions between two documents to magnify the relevance signals (Hofstätter et al., 2019; Hu et al., 2014; Pang et al., 2016; Hui et al., 2017). For example, DRMM (Guo et al., 2016) ranks documents based on the matching histogram of each query and document. Conv-KNRM (Dai et al., 2018) extends DRMM by pairwise n-gram similarity. FastText+ConvKNRM (Hofstätter et al., 2019) further makes use of subword-token embeddings to tackle the vocabulary mismatch problem. Wang et al. (2017) introduce a novel attention-based representation approach to leverage information aggregated

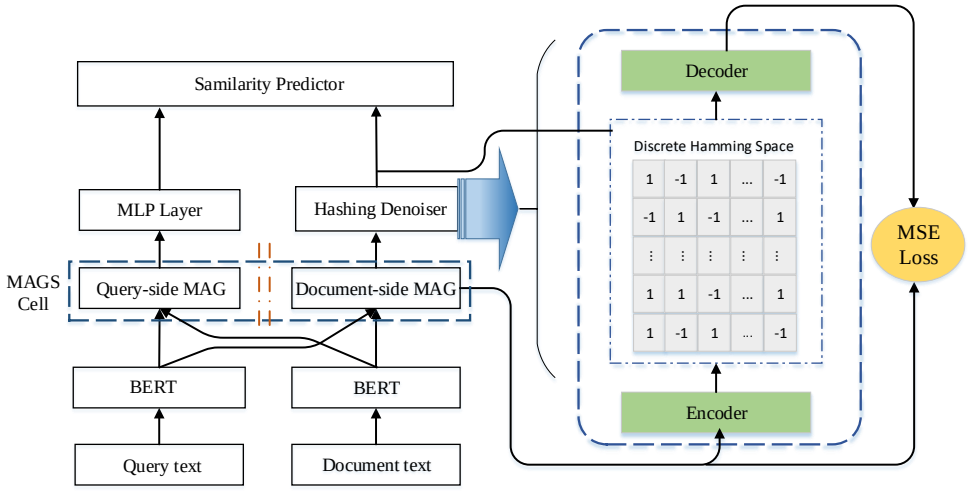


Figure 2: The whole network architecture of ADDAX.

from both question and passage for better predict answers.

2.2 Deep LM Based Matcher

The pre-training language representation models (PLMs), like BERT (Devlin et al., 2019), Roberta (Liu et al., 2019) and XLnet (Yang et al., 2019), have shown great capacity in encoding contextual information. It significantly outperforms other CNN-based and RNN-based neural models in many text matching tasks (Nogueira et al., 2019b,a; Karpukhin et al., 2020). DeepCT (Dai and Callan, 2019a) utilizes a deep contextualized term weighting framework for document retrieval, where contextualized text representations produced by PLMs are used to derive the context-aware term importance weights.

ColBERT (Khattab and Zaharia, 2020) performs interaction MaxSim operator on each PLMs-based word embedding in the query/document to calculate matching scores. Later, TCT-ColBERT (Lin et al., 2020) utilizes knowledge distillation to enhance query latency and greatly reduce the memory cost of ColBERT. CLEAR (Gao et al., 2021b) exploits a neural embedding matching model as a supplement towards conventional lexical matching. COIL (Gao et al., 2021a) utilizes vector similarities between query-document overlapping term contextualized representations for efficient search. Sun et al. (2021) combine the latent topic of the document with its PLMs-based representations to predict the relevance of documents given a query.

Comparing with these existing solutions, our proposed ADDAX goes a step further to identify discriminative features and perform feature denoising by considering the asymmetric nature of many text matching tasks.

2.3 Semantics-Preserving Hashing

Semantics-preserving hashing (Salakhutdinov and Hinton, 2009; Li et al., 2016) aims to learn a concise representation of the input by preserving the core semantics, which can be considered as a form of loss-free dimension reduction. Following this idea, VDSH (Chaidaroon and Fang, 2017) derives hashing codes with variational autoencoders (VAE) to reconstruct and preserve the semantics of the original text. NASH (Shen et al., 2018) learns a VAE-based generative model whose input are TF-IDF vectors, which treats binary codes as Bernoulli latent-variables. Here, we utilize an locality-constrainted hashing to filter out irrelevant information for the long document of a text pair, which can enable more precise text matching.

3 The proposed method

In this section, we first describe the task formulation in Section 3.1. Afterwards, we present each component of ADDAX in detail.

3.1 Task Formulation

Without loss of generality, we assume there are a short query Q and a long document D in an asymmetric text matching pair: $Q = \{q_1, \cdots, q_l\}$ and $D = \{d_1, \cdots, d_t\}$, where $l \ll t$. Here, q_i and d_j indicate the i-th and j-th token in the sequences respectively, and l and t are the number of tokens in the sequences respectively. The goal of the asymmetric matching $f(Q, D)$ is to predict whether Q and D hold a target relation r, where $r \in \{0, 1\}$.

3.2 Architecture

The network architecture of our proposed ADDAX is shown in Figure 2. The entire framework consists of four main parts: a BERT-based context encoder, a matching-adapted gating siamese cell, a locality-constrained hashing denoiser and a MaxSim (Khattab and Zaharia, 2020) based relevance predictor.

BERT-based Context Encoder. We choose to utilize BERT[1] as our context encoder. The BERT-based context encoder can be described as follows:

$$\mathbf{U}_Q = BERT([CLS]q_1q_2\cdots q_l) \quad (1)$$

$$\mathbf{V}_D = BERT([CLS]d_1d_2\cdots d_t) \quad (2)$$

where $\mathbf{U}_Q \in \mathbb{R}^{l\times d}$ and $\mathbf{V}_D \in \mathbb{R}^{t\times d}$ are word-level contextual representations derived for query Q and document D respectively. Parameter d denotes the output dimension of BERT, and `[CLS]` is a specific token indicating the beginning of the token sequence. To reduce the total number of parameters in ADDAX, mitigate overfitting, and facilitate feature interactions across the two texts, we share a single context encoder for both Q and D.

Matching-Adapted Gating Siamese Cell. A human being can identify the relation between two sequences (*e.g.,* query-document, keyword-document, and question-answer) at a glance. For instance, a well-trained graduate student can easily categorize the papers in his/her research direction in term of title and abstract, because he/she can subconsciously identify the discriminative features, and ignore the irrelevant features for the decision.

Here, we simulate this feature discrimination process with a matching-adapted gating siamese cell (called MAGS). It is a parallel architecture with two MAG cells, namely the query-side MAG and the document-side MAG (ref. Figure 3). Since the both query-side and document-side MAGs are identical (but with different parameters), we mainly describe the query-side MAG for simplicity.

Given $\mathbf{U}_Q = [\mathbf{u}_1;\cdots;\mathbf{u}_l]$ and $\mathbf{V}_D = [\mathbf{v}_1;\cdots;\mathbf{v}_t]$ derived by the context encoder, we aim to identify discriminative features and composite them as relevance features. At first, a cross-attention mechanism is utilized to calculate the word-level similarity as follows:

$$\mathbf{S} = \mathbf{U}_Q\mathbf{V}_D^\top \quad (3)$$

where $\mathbf{S} \in \mathbb{R}^{l\times t}$ is the similarity matrix for all the word pairs across the two texts. We then normalize these similarity scores and derive a reference representation in terms of $\mathbf{V}_D$ for each word in Q:

$$\mathbf{R}_Q = softmax(\mathbf{S})\mathbf{V}_D \quad (4)$$

where $softmax$ function is applied over each row of $\mathbf{S}$, and i-th row of $\mathbf{R}_Q$ is the reference representation for i-th word in Q. The purpose of this step is to perform soft feature selection from $\mathbf{V}_D$ according to $\mathbf{S}$. That is, the relevant information in D is *transferred* to represent Q.

However, during this reference representation process, irrelevant information in Q is also presented for further relevence learning. Hence, we construct supplementary features by considering the divergence of the reference representations against the original ones: $\mathbf{D}_Q = \mathbf{U}_Q - \mathbf{R}_Q$, which works as another form of semantic signals. Note that, the above cross-attention mechanism just blindly searches for the most similar token in D to reconstruct Q despite the fact that the most similar one in D might be an meaningless match. Thus, we choose to leverage the attention patterns expressed in $\mathbf{S}$ to both identify the importance of $\mathbf{D}_Q$ and $\mathbf{R}_Q$ as well as the importance of each individual feature in them:

$$\mathbf{E} = \sigma(\mathbf{S}\mathbf{W}_1 + \mathbf{B}_1) \quad (5)$$

$$\mathbf{F}^{(r)} = \mathbf{R}_Q \odot \mathbf{E} \quad (6)$$

$$\mathbf{F}^{(d)} = \mathbf{D}_Q \odot (\mathbf{1} - \mathbf{E}) \quad (7)$$

$$p_i = \sigma(\mathbf{S}_i\mathbf{w}_1 + b_1) \quad (8)$$

$$\mathbf{F}_i^{(c)} = p_i \cdot \mathbf{F}_i^{(r)} \oplus (1 - p_i) \cdot \mathbf{F}_i^{(d)} \quad (9)$$

where $\sigma(\cdot)$ denotes the *sigmoid* function, $\mathbf{W}_1, \mathbf{B}_1 \in \mathbb{R}^{t\times d}$, $\mathbf{w}_1 \in \mathbb{R}^{t\times 1}$ and $\mathbf{b}_1$ are learnable matrices and the bias, $\mathbf{S}_i$, $\mathbf{F}_i^{(c)}$, $\mathbf{F}_i^{(r)}$ and $\mathbf{F}_i^{(d)}$, subscript i indicates i-th row of the corresponding matrix respectively, $\odot$ and $\oplus$ are the element-wise product and vector concatenation operation respectively.

Afterwards, we adopt a highway network to generate the discriminative features $\mathbf{h}_i^Q$ for each word in Q:

$$\mathbf{p}_i = \text{relu}\left(\mathbf{W}_3\mathbf{F}_i^{(c)} + \mathbf{b}_3\right) \quad (10)$$

$$\mathbf{g}_i = \text{sigmoid}\left(\mathbf{W}_4\mathbf{F}_i^{(c)} + \mathbf{b}_4\right) \quad (11)$$

$$\mathbf{i}_i = (1 - \mathbf{g}_i) \odot \mathbf{F}_i^{(c)} + \mathbf{g}_i \odot \mathbf{p}_i \quad (12)$$

$$\mathbf{h}_i^Q = \mathbf{W}_5\mathbf{i}_i + \mathbf{b}_5 \quad (13)$$

[1] We used the base, uncased variant of BERT.

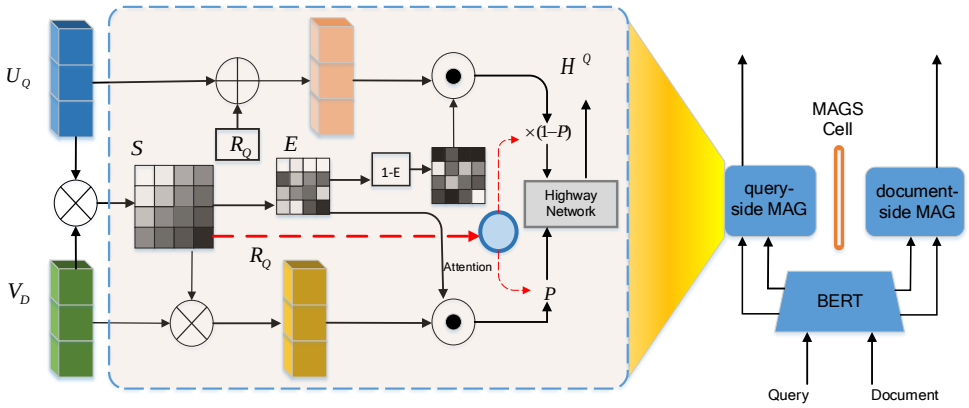


Figure 3: The network structure of MAGS.

where $\mathbf{W}_3, \mathbf{W}_4 \in \mathbb{R}^{2d\times 2d}$ and $\mathbf{W}_5 \in \mathbb{R}^{d\times 2d}$ are the transformation matrices, $\mathbf{b}_3$, $\mathbf{b}_4$ and $\mathbf{b}_5$ are bias vectors. We can form the resultant hybrid discriminative features as a matrix: $\mathbf{H}^Q = [\mathbf{h}_1^Q; \cdots; \mathbf{h}_l^Q]$.

Similarly, the document-side MAG cell switches the roles of Q and D for the same process, but with different parameters. The discriminative features are denoted as $\mathbf{H}^D = [\mathbf{h}_1^D; \cdots; \mathbf{h}_t^D]$.

Locality-Constrained Hashing Denoiser. Since document D is much larger than query Q, the discriminative feature extraction performed by the document-side MAG could still introduce many semantic noises. Here, we adopt a locality-constrained hashing denoiser to further filter out irrelevant features. More specifically, the locality-constrained hashing denoiser defines an encoding function $\mathcal{F}_{en}$, a hashing function $\mathcal{F}_h$, and a decoding function $\mathcal{F}_{de}$.

Encoder function $\mathcal{F}_{en}$ maps the representations $\mathbf{H}^D$ into a low-dimensional matrix $\mathbf{B} \in \mathbb{R}^{t\times h}$. Here, we model $\mathcal{F}_{en}$ as a feed forward network ($FNN(\cdot)$) implemented by a three-layer multi-layer perceptron (MLP), where the $ReLU(\cdot)$ is utilized as the activation function in the second layer to skip unnecessary features and retain discriminating clues (others are $tanh(\cdot)$). The encoding process can be summarized as: $\mathbf{B} = \mathcal{F}_{en}(\mathbf{H}^D) = FFN(\mathbf{H}^D)$.

Hashing function $\mathcal{F}_h$ is devised to learn discriminative binary features for purification and efficient matching. Generally, the $sgn(\cdot)$ function is the best choice for binarization, but not differentiable. Hence, we use an approximate function $tanh(\cdot)$ to replace $sgn(\cdot)$ for supporting model training. Specifically, the hashing function is written as: $\mathbf{B}^D = \mathcal{F}_h(\mathbf{B}) = \tanh(\alpha\mathbf{B})$. The hyper-parameters α is a coefficient helping generate balanced and discriminative hash codes. To ensure that the values in $\mathbf{B}^D \in \{-1, 1\}$, we define an extra constraint (*MSE* loss) (Li et al., 2016; Xu and Li, 2020): $\mathcal{L}_1 = \left\|\mathbf{B}^D - \boldsymbol{\mathcal{B}}^{(b)}\right\|_F^2$, where $\boldsymbol{\mathcal{B}}^{(b)} = sgn(\mathbf{B})$, and $\|\cdot\|_F$ is the Frobenius norm.

Similar to $\mathcal{F}_{en}$, decoding function $\mathcal{F}_{de}$ recovers $\mathbf{H}^D$ from $\mathbf{B}^D$ with a three-layer MLP (encoder transpose). Hence, the reconstructed matrix $\mathbf{H}_r^D \in \mathbb{R}^{t\times d}$ can be written as: $\mathbf{H}_r^D = \mathcal{F}_{de}(\mathbf{B}^D) = FFN^\top(\mathbf{B}^D)$. To preserve the core semantics during the reconstruction, a *MSE* loss is used to guide the model training: $\mathcal{L}_2 = \left\|\mathbf{H}_r^D - \mathbf{H}^D\right\|_F^2$.

We can also perform a hashing denoiser for $\mathbf{H}^Q$. However, we did not observe improvement due to noiseless nature of a short query. Instead, the matrix representations $\mathbf{H}^Q$ of query Q are updated with a single MLP layer to match the dimension of the hashing denoiser: $\mathbf{H}^Q = MLP(\mathbf{H}^Q)$, where $\mathbf{H}^Q \in R^{l\times h}$ is used for final prediction.

Similarity Predictor. With both $\mathbf{H}^Q = [\mathbf{h}_1^Q; \cdots; \mathbf{h}_l^Q]$ and $\mathbf{B}^D = [\mathbf{b}_1^D; \cdots; \mathbf{b}_t^D]$, the matching score between Q and D, $f(Q, D)$, is estimated via a MaxSim operator (Khattab and Zaharia, 2020) as follows:

$$f(Q, D) = \sum_i^l \max_j^t \cos(\mathbf{h}_i^Q \cdot \mathbf{b}_j^D) \qquad (14)$$

where function $cos(\cdot)$ calculates the cosine similarity of the given vectors.

3.3 Model Optimization

The objective of model optimization is to guide the relevance learning of ADDAX and help estimate the matching score of the asymmetric text pair. During the training stage, we uti-

lize the negative sampling strategy via a triplet-based hinge loss (Xu and Li, 2020): $\mathcal{L}_3 = \max\{0, 1.0 - f(Q, D) + f(Q, D^-)\}$, where D^- is the corresponding negative document sampled from the training set.

Finally, we need to combine the hinge loss and two constraints in hashing denoiser together. That is, the final optimization objective for ADDAX is a linear fusion of $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$:

$$\min_{\theta} \mathcal{L} = \sum_{(Q,D,D^-)} [\mathcal{L}_3 + \delta \cdot \mathcal{L}_1 + \gamma \cdot \mathcal{L}_2] \quad (15)$$

where δ and γ are tunable hyper-parameters controling the importance of each constraint respectively, θ is the parameter set of ADDAX. We use Adam (Kingma and Ba, 2014) for parameter update in an end-to-end fashion over mini-batches.

4 EXPERIMENTS

4.1 Experimental Settings

Datasets. Our experiments are conducted on four real-world datasets, covering the tasks of both question answer and document retrieval: *InsuranceQA* (Feng et al., 2015) is a widely used benchmark for QA. We leverage the v1.0 version of this corpus. *WikiQA* (Yang et al., 2015) is an open-domain answer selection dataset. We follow the preprocessing utilized in (Xu and Li, 2020) to filter out the questions that have no positive answers; *YahooQA*[2] is a collection constructed from Yahoo! Answers. In order to ensure that it has sufficient asymmetric text pairs, sentences with length among the range of 16 - 24 are filtered; *MS MARCO*[3] is a benchmark for information retrieval. It is a collection of 8.8M passages from web pages and contains approximately 400M tuples of a query, positive and negative passages. We utilize the provided data partition for model training and evaluation.

The average length ratio for a query and a document is greater than 3 for these datasets, which conform to asymmetric text matching scenarios.

Baselines. We compare our proposed ADDAX with two types of state-of-the-art baselines. The first type can perform question-answer (QA) matching. The chosen baseline models for answer selection can be partitioned into four categories: **(a)** *conventional single models*: IARNN-GATE (2016), AP-CNN (2016), RNN-POA (2017), AP-BiLSTM (2016), HD-LSTM (2017), AP-LSTM (2018), Multihop-Sequential-LSTM (2018), HyperQA (2018), MULT (2016), TFM+HN (2019), LSTM-CNN+HN (2019); **(b)** *single models that exploit external knowledge*: KAN (2018), CK-ANN (2021), CKANN-L (2021); **(c)** *ensemble models*: SUM$_{BASE,PTK}$ (2018), LRXNET (2018), SD (BiLSIM+TFM) (2020); **(d)** *BERT-based models*: HAS (2020), DDR-Match(BERT,WD) (2022) and BERT$_{base}$ is implemented by ourselves.

As to document retrieval, we include the following methods for comparison: BM25 (2018), PACRR (2017), KNRM (2017) and fastText+ConvKNRM (2019). In addition, since the proposed ADDAX adopts the BERT as the context encoder, we then pick several uptodate LMs-based models, including BERT$_{base}$ ranker (2019), DeepCT (2019a), docT5query (2019a), ColBERT (2020), TCT-ColBERT (2020), COIL-tok (2021a) and COIL-full (2021a). Furthermore, we also include two dense retrievers for performance comparison, *i.e.,* RepCONC (2022), CLEAR (2021b) and ADORE+STAR (2021).

Parameter Settings and Evaluation Metrics. In our experiments, we choose $BERT_{base}$ as the context encoder in ADDAX. To be more specific, we set the hidden dimension h to be 300. The mini-batch size for insuranceQA, wikiQA, yahooQA, and MS MARCO is set to be 32, 32, 64, and 64, respectively. The probability of dropout is set to be 0.1. The learning rate for insuranceQA, MS MARCO, wikiQA and yahooQA is $5e^{-6}$, $5e^{-6}$, $1e^{-5}$, and $9e^{-6}$, respectively. The numbers of training epoches are 60 for insuranceQA, 18 for wikiQA and 9 for yahooQA. In addition, we train ADDAX for 200k iterations for MS MARCO. The values of α, δ and γ are set to 5, $1e^{-6}$, 0.003 respectively. To enable fair comparison, we choose the common evaluaton metrics utilized in these baselines and directly reuse the reported results from the corresponding papers.

4.2 Performance Comparison

Table 1 summarizes the performance of 22 methods for answer selection on the corresponding three datasets. We choose to discuss the experimental results on each dataset separately.

On InsuranceQA, We can observe that conventional single models like IARNN-GATE, MULT, and TFM+HN perform much better than other sin-

[2]https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&guccounter=1

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

Table 1: Performance comparison on the QA datasets (best in boldface). Results not applicable and not available are denoted "–" and "n.a." respectively. HAS-HL represents a model variant of HAS without a hashing layer. Significant improvement with respect to HAS is indicated (†) (p-value ≤ 0.05).

	insuranceQA		wikiQA		yahooQA	
Model	P@1(Test1)	P@1(Test2)	MAP	MRR	P@1	MRR
IARNN-GATE	70.10	62.80	72.58	73.94	–	–
AP-CNN	69.80	66.30	68.86	69.57	56.00	72.60
AP-BiLSTM	71.70	66.40	67.05	68.42	56.80	73.10
HD-LSTM	–	–	–	–	55.70	73.50
HyperQA	n.a.	n.a.	71.20	72.70	68.30	80.10
RNN-POA	n.a.	n.a.	72.12	73.12	n.a.	n.a.
Multihop-Sequential-LSTM	70.50	66.90	72.20	73.80	n.a.	n.a.
AP-LSTM	69.00	64.80	68.90	69.60	n.a.	n.a
MULT	75.20	73.40	74.33	75.45	n.a.	n.a.
LSTM-CNN+HN	73.30	69.10	–	–	–	–
TFM+HN	75.60	73.40	–	–	–	–
KAN (Tgt-Only)	71.50	68.80	–	–	67.20	80.30
KAN	75.20	72.50	–	–	74.40	84.00
CKANN	76.30	**75.10**	73.20	75.50	84.40	90.20
CKANN-L	75.90	74.90	72.80	73.90	84.20	90.60
$\text{SUM}_{BASE,PTK}$	–	–	75.59	77.00	–	–
LRXNET	–	–	76.57	75.10	–	–
SD (BiLSIM+TFM)	–	–	70.40	71.20	–	–
BERT_{base}	74.52	71.97	75.30	77.00	73.49	81.93
DDR-Match(BERT,WD)	n.a.	n.a.	79.58	81.23	n.a.	n.a.
HAS	76.38	73.71	81.01	82.22	73.89	82.10
HAS-HL	76.12	74.12	80.65	81.83	74.78	82.68
ADDAX	**77.83**†	74.83†	**82.50**†	**83.38**†	**87.63**	**90.69**

gle models. Also, it is not surprising that the BERT-based methods (*e.g.,* HAS) consistently yield the better performance compared to single models. This is expected since the LMs can absorb large-scale common knowledge to help bridge the vocabulary mismatch. These observations are consistent with many previous works (Xu and Li, 2020). Single models that exploit external knowledge (*e.g.,* KAN and CKANN) are superior to those convetional single models and BERT-based models, mainly because the external knowledge is very helpful. As a comparison, our ADDAX achieves significantly better performance than almost all baselines in InsuranceQA dataset (except for CKANN on Test2 set).

On WikiQA, it is surprising that single models exploiting external knowledge can not obtain obvious advantages compared to some single models. The possible reasons for this phenomenon could be the scarity of the training data and irrelevant external knowledge. Secondly, ensemble models like $\text{SUM}_{BASE,PTK}$ and LRXNET significantly outperform SD (BiLSIM+TFM). Also, the ensemble models obtain substantial performance gain than the both conventional single models and the ones with external knowledge, indicating effectiveness of integrating multiple models in improving the generalization ability. Thirdly, BERT_{base} consistently performs worse than HAS-HL and HAS. This observation is consistent across all the four datasets, suggesting positive benefit of model feature interactions. As to ADDAX, a much better performance is obtained against all baselines here.

On YahooQA, we observe a similar performance pattern as with the InsuranceQA dataset. Our proposed ADDAX substantially outperforms all baselines in terms of P@1 and MRR. Specifically, compared with the best baseline, our ADDAX obtains relative $P@1$ gain of 3.23%.

Table 2 reports the performance comparison of different document retrieval models on MS MARCO. For the neural matching models, LMs-based methods obtain much better performance than PACRR, KNRM and fastText+ConvKNRM, suggesting the powerful language expression ability of the former. Note that DeepCT and DocT5Query can adaptively adjust the term importance by exploiting LMs, but they are still inferior in semantic matching. Also, it is worth noting that dense retrievers are almost on par with the LMs-based models. In contrast, ADDAX consistently achieves the best performance on MS MARCO

Table 2: Results on MS MARCO (best in boldface).

Model	MS MARCO MRR@10(dev)
BM25	18.70
KNRM	19.80
PACRR	25.90
fastText+ConvKNRM	29.00
BERT_{base}	34.70
DeepCT	24.30
docT5query	27.70
ColBERT	34.90
TCT-ColBERT	33.50
COIL-tok	33.60
COIL-full	34.80
CLEAR	33.80
RepCONC	34.00
ADORE+STAR	34.70
ADDAX	**36.15**

dataset. Specifically, the performance gain by ADDAX over all the baselines is in the range of 1.25%-17.40% in terms of MRR@10.

Overall, the above comparisons made over two different tasks consistently show that the proposed ADDAX achieves substantial performance gain in general. These promising results validate that the matching-adapted gating siamese cell and the hashing denoiser proposed in ADDAX are effective in performing feature discrimination and denoising for asymmetric text matching.

4.3 Model Analysis

Ablation Study. Here, we perform a series of ablation studies to explore how each design in ADDAX affects the asymmetric text matching. To be more specific, we compare ADDAX with the following variants: (a) **w/o MAGS**, removing the matching-adapted gating siamese cell; (b) **w/o HW**, eliminating highway network to fuse the two kinds of semantic signals, but add them directly; (c) **w/o HD**, excluding the locality-constrained hashing denoiser; (d) **Att-MAGS**, in the case of w/o HD, keeping only cross-attention mechanism in MAGS.

Table 3 reports the results on MS MARCO and wikiQA datasets. We can see that the exclusion of the matching-adapted gating siamese cell incurs the largest performance degradation, followed by the hashing denoiser. Particularly, w/o MAGS drops absolutely by 5.30% and 4.61% on wikiQA in terms of MAP and MRR, respectively, and 1.25% on MS MARCO in terms of MRR@10. In addition, Att-MAGS is also worse than MAGS. These demonstrates that the matching-adapted gating siamese cell is effective in identifing discriminative features to enhance matching accuracy. Besides, w/o HD also results in worse performance, suggesting the effectiveness of performing feature-level denoising on document side. For example, we illustrate the feature heatmap of a random sample $\mathbf{B}^D$ generated by ADDAX and w/o HD (*i.e.,* as shown in Figure 4(d)). We can see that the denoiser indeed filters many features. For each specific structure designed in MAGS, we also make the following observation: the performance drop of w/o HW suggests that the highway network is more effective to composite the hybrid discriminative features.

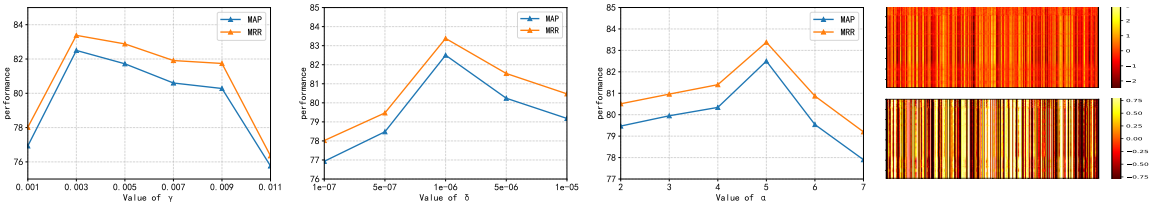
Table 3: The performance of different ADDAX variants on wikiQA and MS MARCO (best in boldface).

Model	MS MARCO MRR@10(dev)	wikiQA MAP	wikiQA MRR
w/o MAGS	34.90	77.20	78.77
w/o HW	35.32	79.01	80.23
w/o HD	35.49	80.15	81.58
Att-MAGS	35.00	79.79	81.34
ADDAX	**36.15**	**82.50**	**83.38**

In general, our proposed ADDAX consistently surpasses five variants on MS MARCO and wikiQA datasets, demonstrating the validity of each component design.

Sensitivity Analysis of α, δ and γ. We further investigate the sensitivity of hyper-parameters (*i.e.,* α, δ and γ) in ADDAX on the wikiQA test set. Recall that δ controls the importance of the constraint loss $\mathcal{L}_1$, γ controls the importance of hashing denoiser's reconstruction loss $\mathcal{L}_2$, and α control the balance of the hash codes. When studying a parameter, the other two parameters are fixed to the values described in Section 4.1.

From Figure 4(c), we can see that the matching performance starts growing by increasing α to 5. Moreover, Figure 4(b) plots the performance pattern by varying δ values. We observe that ADDAX is not sensitive to δ in the range of $[1e^{-6}, 5e^{-6}]$ and obtains better performance at $\delta = 1e^{-6}$. Figure 4(a) plots the performance pattern by varying γ values. When γ is greater than or less than 0.003, the performance becomes much worse.



(a) Sensitivity of γ. (b) Sensitivity of δ. (c) Sensitivity of α. (d) Feature heatmap.

Figure 4: Performance with varying parameters on wikiQA dataset. The upper and lower subgraphs in Figure (d) represent feature heatmap of w/o HD and ADDAX on $\mathbf{B}^D$ respectively.

5 CONCLUSION

In this paper, we introduce an adaptive feature discrimination and denoising model for asymmetric text matching. Specifically, we first design a matching-adapted gating siamese cell in ADDAX to perform feature discrimination and generate the hybrid representations together for the asymmetric text pair. We then present a locality-constrained hashing denoiser for filtering semantic noise for redundant long documents. Extensive experimental results on four benchmarks have demonstrated the effectiveness and superiority of our proposed ADDAX. As future work, we plan to investigate the possibility of feature discrimination and denoising in other asymmetric scenarios like document abstractive summarization, caption generation and more.

Acknowledgements

This work was supported by National Key Research and Development Program of China (No. 2020AAA0107904). National Natural Science Foundation of China (No. U21B2027, 61866020), Key Research and Development Program of Yunnan Province (No. 202103AA080015). Natural Science Foundation Project of Yunnan Science and Technology Department (No. 2019FB082).

References

Suthee Chaidaroon and Yi Fang. 2017. Variational deep semantic hashing for text documents. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 75–84.

Dong Chen, Shaoliang Peng, Kenli Li, Ying Xu, Jinling Zhang, and Xiaolan Xie. 2020. *Re-Ranking Answer Selection with Similarity Aggregation*, page 1677–1680. Association for Computing Machinery, New York, NY, USA.

Qin Chen, Qinmin Hu, Jimmy Xiangji Huang, Liang He, and Weijie An. 2017. Enhancing recurrent neural networks with positional attention for question answering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–996.

Zhuyun Dai and Jamie Callan. 2019a. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*.

Zhuyun Dai and Jamie Callan. 2019b. Deeper text understanding for IR with contextual neural language modeling. In *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 126–134.

Yang Deng, Ying Shen, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. 2018. Knowledge as a bridge: Improving cross-domain answer selection with external knowledge. In *Proceedings of the 27th international conference on computational linguistics*, pages 3295–3305.

Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Wai Lam, and Ying Shen. 2021. Contextualized knowledge-aware attentive neural network: Enhancing answer selection with knowledge. *ACM Transactions on Information Systems*.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820. IEEE.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021b. Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval–43rd European Conference on IR Research*.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for Ad-hoc retrieval. In *International Conference on Information and Knowledge Management, Proceedings*.

Sebastian Hofstätter, Navid Rekabsaz, Carsten Eickhoff, and Allan Hanbury. 2019. On the effect of low-frequency terms on neural-IR models. In *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard De Melo. 2017. Pacrr: A position-aware neural ir model for relevance matching. *arXiv preprint arXiv:1704.03940*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sawan Kumar, Shweta Garg, Kartik Mehta, and Nikhil Rasiwasia. 2019. Improving answer selection and answer triggering using hard negatives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5911–5917, Hong Kong, China. Association for Computational Linguistics.

Wu Jun Li, Sheng Wang, and Wang Cheng Kang. 2016. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI International Joint Conference on Artificial Intelligence*.

S. C. Lin, J. H. Yang, and J. Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers.

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

Shashi Narayan, Ronald Cardenas, Nikos Papasarantopoulos, Shay B Cohen, Mirella Lapata, Jiangsheng Yu, and Yi Chang. 2018. Document modeling with external attention for sentence extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2020–2030.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019a. From doc2query to doctttttquery. *Online preprint*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019b. Multi-stage document ranking with BERT.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Dinghan Shen, Qinliang Su, Paidamoyo Chapfuwa, Wenlin Wang, Guoyin Wang, Lawrence Carin, and Ricardo Henao. 2018. Nash: Toward end-to-end neural architecture for generative semantic hashing. *arXiv preprint arXiv:1805.05361*.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. 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*, pages 101–110.

Xingwu Sun, Yanling Cui, Hongyin Tang, Qiuyu Zhu, Fuzheng Zhang, and Beihong Jin. 2021. Tita: A two-stage interaction and topic-aware text matching model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5431–5440.

Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Learning to rank question answer pairs with holographic dual lstm architecture. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 695–704.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Cross temporal recurrent networks for ranking question answer pairs. In *Thirty-second AAAI conference on artificial intelligence*.

Nam Khanh Tran and Claudia Niedereée. 2018. Multihop attention networks for question answer matching. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 325–334.

Kateryna Tymoshenko and Alessandro Moschitti. 2018. Cross-pair text representations for answer sentence selection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2162–2173.

Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1288–1297.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-To-end neural ad-hoc ranking with kernel pooling. In *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Dong Xu and Wu-Jun Li. 2020. Hashing based answer selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9330–9337.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using lucene. *Journal of Data and Information Quality (JDIQ)*, 10(4):1–20.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Weijie Yu, Chen Xu, Jun Xu, Liang Pang, and Ji-Rong Wen. 2022. Distribution distance regularized sequence representation for text matching in asymmetrical domains. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:721–733.

Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *International Conference on Information and Knowledge Management, Proceedings*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. *arXiv preprint arXiv:2104.08051*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Learning discrete representations via constrained clustering for effective and efficient dense retrieval. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1328–1336.