# Topic-enhanced knowledge-aware retrieval model for diverse relevance estimation

Xiangsheng Li[1], Jiaxin Mao[2], Weizhi Ma[1], Yiqun Liu[1*], Min Zhang[1], Shaoping Ma[1], Zhaowei Wang[3] and Xiuqiang He[3]
[1]Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China
[3]Noah's Ark Lab, Huawei
yiqunliu@tsinghua.edu.cn

## ABSTRACT

Relevance measures the relation between query and document which contains several different dimensions, e.g., semantic similarity, topical relatedness, cognitive relevance (the relations in the aspect of knowledge), usefulness, timeliness, utility and so on. However, existing retrieval models mainly focus on semantic similarity and cognitive relevance while ignore other possible dimensions to model relevance. Topical relatedness, as an important dimension to measure relevance, is not well studied in existing neural information retrieval. In this paper, we propose a Topic Enhanced Knowledge-aware retrieval Model (TEKM) that jointly learns semantic similarity, knowledge relevance and topical relatedness to estimate relevance between query and document. We first construct a neural topic model to learn topical information and generate topic embeddings of a query. Then we combine the topic embeddings with a knowledge-aware retrieval model to estimate different dimensions of relevance. Specifically, we exploit kernel pooling to soft match topic embeddings with word and entity in a unified embedding space to generate fine-grained topical relatedness. The whole model is trained in an end-to-end manner. Experiments on a large-scale publicly available benchmark dataset show that TEKM outperforms existing retrieval models. Further analysis also shows how topic relatedness is modeled to improve traditional retrieval model with semantic similarity and knowledge relevance.

## KEYWORDS

Neural IR; Neural topic model; Knowledge graph; Kernel pooling

**ACM Reference Format:**
Xiangsheng Li, Jiaxin Mao, Weizhi Ma, Yiqun Liu, Min Zhang, Shaoping Ma, Zhaowei Wang and Xiuqiang He. 2021. Topic-enhanced knowledge-aware retrieval model for diverse relevance estimation. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449943

*Corresponding author.


*WWW '21, April 19–23, 2021, Ljubljana, Slovenia*

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449943

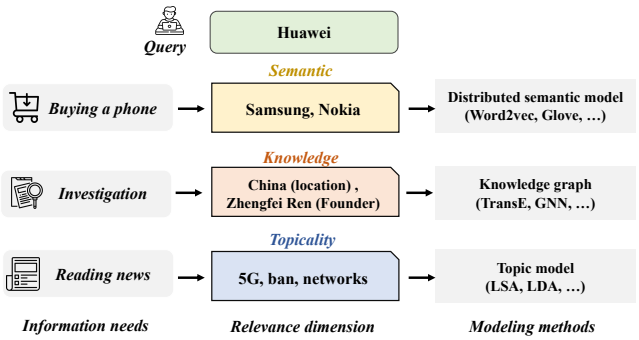



**Figure 1: Different relevance dimensions for the query "Huawei".**

## 1 INTRODUCTION

Relevance estimation is a central problem in information retrieval (IR) research, which aims to learn a scoring function to capture the relevance of a document with respect to a query. Intuitively, relevance implicitly or explicitly involves a relation. It's a measure for determining a degree of appropriateness or effectiveness with respect to *the matter at hand* [35]. Priors studies believed that relevance potentially has different manifestations [2, 33, 34]. Generally, these manifestations address that the relations behind relevance should involve different dimensions to measure, e.g., semantic similarity, topical relatedness, cognitive relevance, usefulness, utility to a situation or problem at hand, temporal aspect, intent in use and so on [35]. These dimensions provide a base or context for establishing a relation, which can be explicit or implicit, well-defined or visceral. To build a better IR system, different dimensions of relevance needs to be considered so that IR system can better infer user search intent and provide a more accurate estimation of relevance score.

Among these dimensions, semantic similarity, cognitive relevance (the relations in the aspect of knowledge) and topical relatedness draw the most attention in IR research [27] since they are relatively explicit and well-defined compared with other dimensions. We take an example query "*Huawei*" to understand their differences in Figure 1. Cognitive relevance (or knowledge relevance) [35] is the cognitive correspondence or informativeness between objects, e.g., *Huawei* - (*China*, *Zhengfei Ren*). In IR literature, we often estimate knowledge relevance by external knowledge bases. Difference between semantic similarity and topical relatedness is easier to

understand if we look back upon their modeling methods. Semantic similarity is often modeled by distributed semantic models like word2vec [25] while topic relatedness is often modeled by topic models like LDA [3]. The former is learned within a local context (or $n$-gram windows) while the latter takes the whole document as context. Difference in their considered contextual information leads to the capture of different types of similarity. Local context-based models (word2vec) capture semantic similarity (e.g. *Huawei* - (*Samsung*, *Nokia*)) while document-based models (LDA) capture semantic relatedness (i.e., topical relatedness) like *Huawei* - (*5G*, *ban*, *networks*). By applying word2vec to test the topical relevant words in Figure 1, we find the similarities are very low, which indicates that traditional distributed word embedding cannot capture the topical relatedness. Therefore, it is necessary to model the topical relatedness to improve the current neural IR solutions.

Recently, neural retrieval models have gained much attention in IR community and obtain promising ranking performance. According to the investigation of neural IR [11] , most existing neural retrieval models are designed to model semantic similarity between query and document based on their distributed word embeddings. These models are able to achieve better ranking performance compared to traditional statistical models like BM25 [31]. To better estimate relevance, further works also attempt to incorporate knowledge graph and use entity embeddings to consider the dimension of knowledge relevance [22, 44]. The combination of semantic similarity and knowledge relevance helps these models better estimate the relevance score and improve ranking performance. Prior studies [17, 23] in traditional retrieval models have shown that topical relatedness is an important dimension for relevance estimation. It provides the signal to measure the relation of two objects under a specific topic. However, to our best knowledge, there is no existing effort in incorporating topical relatedness into the relevance estimation of neural IR models.

In this paper, to improve neural retrieval model by considering different relevance dimensions, we propose a Topic Enhanced Knowledge-aware retrieval Model (TEKM) that jointly learns semantic similarity, knowledge relevance and topical relatedness between query and document. Firstly, we develop a new neural topic model for topic inference. The model is based on variational autoencoder (VAE) [18] for discovering topics via neural networks. Compared to traditional topic model like LDA [3], it is more compatible with other neural models [24] and also enables the whole model to train in an end-to-end manner. Secondly, each topic is represented by an implicit topic vector and topical relatedness is learned by extracting multi-level soft matching signals between topic vectors of a query and textual information of a document. In particular, the textual information is represented by both word and entity annotations. Word embedding [25] inherits context semantic while entity embedding inherits the relations in knowledge graph [22] to neural IR model. Three dimensions of relevance are aggregated by kernel pooling method [43], which models different levels of soft matches. With this framework, the proposed model can jointly learns different dimensions of relevance and estimate a more accurate relevance score.

We conduct experiments on a large-scale public test collection [5] Tiangong-ST from a commercial search engine. Experimental results show that our framework significantly outperforms existing retrieval models. We further investigate the effectiveness of topic information in different ranking scenarios and how topic information improve traditional semantic matching signals for better document ranking. The analysis indicates that by modeling topical relatedness, our model can capture more reliable matching signals compared with traditional neural retrieval models with word and entity interactions.

Our main contributions are three-folds:

(1) We propose a new neural topic model that represents each topic as an implicit topic vector, which is flexible to deep neural network based learning tasks in different application scenarios. In our work, we exploit it to generate topic information for neural information retrieval.
(2) We propose a Topic Enhanced Semantic Matching Model (TESM) that considers different dimensions of relevance estimation. Specifically, our model jointly learns semantic similarity, knowledge relevance and topical relatedness between query and document. To the best of our knowledge, this is the first attempt to integrate topic information into neural retrieval models.
(3) Extensive experiments on a large-scale public test collection show that our framework significantly outperform existing retrieval models. We systematically analyze the effectiveness of topic information in different ranking scenarios, thereby providing a solid understanding of how to effectively utilize topic information with neural retrieval models.

## 2 RELATED WORK

### 2.1 Manifestations of relevance

Relevance is the key notion in information retrieval [19]. It is so basic that people use it without thinking about it. Intuitively, we suppose that relevance always implicitly or explicitly involves a relation. It's a measure for determining a degree of appropriateness or effectiveness with respect to *the matter at hand* [35]. Vickery [39] was first to recognize that relevance has different manifestations, which includes a duality - *user relevance* and *subject relevance*. The former depends on individual user's cognitive state and personalized preference while the latter lies in the content of candidate item itself. Mizzaro [38] further proposed that relevance manifestations can be represented in a four-dimensional space: *information resources* (documents, surrogates, information), *representation of user problem* (real information need, perceived information need, request, query), *time* and *components* (topic, task, context). Those dimensions represent various aspects of user information needs. Further researches followed the problem setting but called these dimensions by a number of different names [2, 33, 34]. In summary, these dimensions include semantic similarity, topical relatedness, cognitive relevance (knowledge relevance), usefulness, utility to a situation or problem at hand, temporal aspect, intent in use and so on [35]. To better meet user's information needs, different dimensions of relevance needs to be taken into consideration so that the IR system can better infer user search intent and provide a more accurate relevance score.

### 2.2 Topic modeling in information retrieval

The earliest method of incorporating topic models in IR was performed by introducing terms from hand-crafted thesauri, which are

typical manually-built topic models. This model incorporated another dimension to measure word similarity but was labor-intensive. Latter, a number of statistical topic models e.g., probabilistic latent semantic analysis (pLSA) [13], latent Dirichlet allocation (LDA) [3] were proposed and widely used in different applications such as emotion classification [21] and recommendation systems [1]. They were able to effectively learn unsupervised representations of text and capture topical features. Recently, neural topic models [24] are proposed based on variational auto-encoder [18], which makes it possible to jointly perform topic inference and other application tasks.

The first LDA-based retrieval model was proposed by Wei et al. [42]. The basic idea is to improve query likelihood function with the maximum likelihood of word $w$ in the document $D$, i.e.,

$$\begin{aligned} P(w|D) &= \lambda P_{QL}(w|D,C) + (1-\lambda)P_{LDA}(w|D) \\ P_{LDA}(w|D) &= \sum_{z} P(w|z)P(z|D) \end{aligned} \tag{1}$$

where $C$ is the collection and $z$ is the topic. The following LDA-based retrieval model is generally based on this framework. Jian et al. [17] improves it by replacing query likelihood with other retrieval models like BM25, MATF [28] and Dirichlet LM [45]. Mendoza et al. [23] propose strategies for topic and model selection to learn better topic information. However, these models are only based on traditional statistical models and are rough combination of two independent models. In neural information retrieval, few investigations have been made to integrate topic information into neural information retrieval. In our work, we attempt to combine topic model with neural retrieval model to better estimate relevance in different dimensions.

## 2.3 Neural Retrieval Models

Existing neural retrieval models can be grouped into two categories [11], namely *representation-based* and *interaction-based* models. Representation-based models aim to build a good semantic representation of queries and documents. DSSM [16] represents two input texts with a unified process by using a multi-layer perceptron (MLP) transformation. ARCI [15] employs convolutional neural networks to replace MLPs to represent the input text. They are computationally efficient but fail to capture fine-grained semantic information (e.g., passage or sentence-level relevance [20]). On the other hand, interaction-based models learn local interaction patterns from query-document pairs and capture more fine-grained semantic information. ARCII [15] utilize convolutional neural networks to capture complicated patterns from word-level interactions. KNRM [43] summarize multi-level soft matches between query and document. Matchpyramid [30] defines a symmetric interaction function to model term similarities between query and document.

These model mainly focus on semantic similarity by generating matching signals with word embedding similarity and ignore other possible dimensions of relevance. To better estimate relevance, a number of previous works also attempt to incorporate knowledge graph to consider the dimension of knowledge relevance (or cognitive relevance). Xiong et al. [44] modeled entity salience in document by modeling the interactions between entities and words. Liu et al. [22] introduced entities in the knowledge base into a neural retrieval model. The interactions in the aspect of knowledge base help the retrieval model perform better relevance estimation in the dimension of cognitive relevance (or knowledge relevance). Considering that topic information includes the signals of a specific component (task, goal, or context) [38] and is different from other relevance dimensions, we attempt to introduce additional dimension on topic information to improve neural retrieval models.

# 3 TOPIC ENHANCED KNOWLEDGE-AWARE RETRIEVAL MODEL

This section presents our proposed Topic Enhanced Knowledge-aware Model (TEKM). The overall framework is shown in Figure 2.

## 3.1 Overview

The intuitive idea behind our model is to incorporate topic embeddings to measure the topical relatedness for relevance estimation, which consists of two components: 1) A neural topic model that learns topic information from the query and generates weighted topic embeddings for further matching. 2) A topic enhanced matching framework that combines semantic similarity, knowledge relevance and topical relatedness for learning to rank.

We first adopt neural topic model instead of traditional topic models like LDA [3] since it benefits in several aspects such as topic quality, computational efficiency and compatibility with neural models [37]. Our neural topic model takes both words and entities as input and represent topics by a set of auxiliary implicit topic embeddings. The embedding representation of topic information helps the fine-grained interactions with both words and entities in the match framework. In addition, we only build topic embeddings for query instead of document. The main idea is to augment the topic representation power of query and use the generated topic embeddings to match topical relevant words and entities in the document (e.g., *Huawei* - (*5G*, *ban*, *networks*) in Figure 1) ). It can be regarded as a query expansion on the topical dimension. The interactions between topic embeddings and word embeddings capture the fine-grained word-level topic matching signals. And it is similar for the relationship between topic embeddings and entity embeddings. We finally exploit kernel pooling to combine fined-grained topic relatedness with semantic similarity and knowledge relevance to estimate the final relevance score. We will explain the model details in the following subsections.

## 3.2 Neural Topic Model

The upper part of Figure 2 shows the structure of our Neural Topic Model (NTM), which is built upon the VAE architecture. The basic idea is to mimic the modeling process of topic models by neural networks. As discussed in Section 3.1, we generate topic information based on the query instead of the document to augment the topic representation power of query. Given a predefined topic number $K$, NTM maps the query to the topic distribution $\boldsymbol{\theta}$ via inference encoder. Then $\boldsymbol{\theta}$ is decoded by the generative decoder to reconstruct the query. Specifically, we represent the query by both words $\boldsymbol{q}_w$ and its entity annotations $\boldsymbol{q}_e$. The generative process of the query is described as follows:

- Draw a topic distribution $\boldsymbol{\theta} \sim Dirichlet(\alpha)$,

Xiangsheng Li[1], Jiaxin Mao[2], Weizhi Ma[1], Yiqun Liu[1*], Min Zhang[1], Shaoping Ma[1], Zhaowei Wang[3] and Xiuqiang He[3]

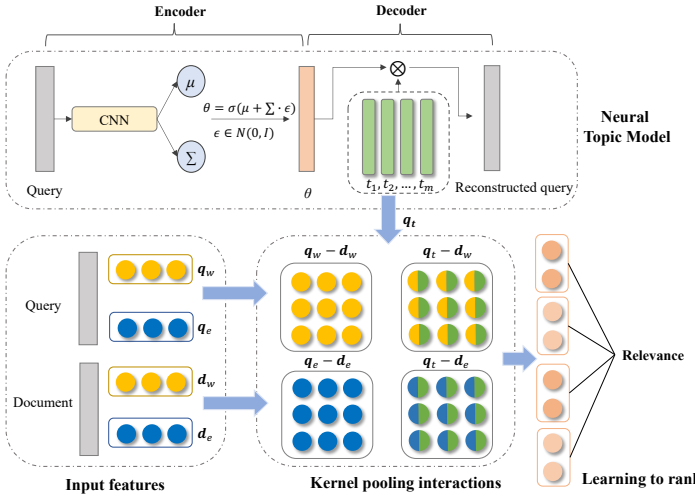



**Figure 2: Framework of the proposed Topic enhanced knowledge-aware retrieval model.**

- For each word $w$ and entity $e$ in the query,
  draw $w \sim Multinomial(1, \sigma(\boldsymbol{\beta\theta}))$,
  draw $e \sim Multinomial(1, \sigma(\boldsymbol{\phi\theta}))$

where $\alpha$ is the hyper-parameter to control the mixture of topics. $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ are the matrices of topic-word probability and topic-entity probability, respectively. $\sigma$ is the softmax function.

The probability of the reconstructed query can thus be represented by summing out the topic distribution $\boldsymbol{\theta}$:

$$\begin{aligned} p(\boldsymbol{q}_w|\alpha, \boldsymbol{\beta}) &= \int_\theta \left( \prod_w^{|\boldsymbol{q}_w|} p(w|\boldsymbol{\beta}, \theta) \right) p(\theta|\alpha) d\theta \\ p(\boldsymbol{q}_e|\alpha, \boldsymbol{\phi}) &= \int_\theta \left( \prod_e^{|\boldsymbol{q}_e|} p(w|\boldsymbol{\phi}, \theta) \right) p(\theta|\alpha) d\theta \end{aligned} \tag{2}$$

where $p(w|\boldsymbol{\beta}, \theta)$ and $p(e|\boldsymbol{\phi}, \theta)$ are both subject to multi-nomial distribution. To make the back-propagated gradient flow through a random node, we should apply reparameterization trick (RT) [32] in the neural networks. However, it is hard to directly develop an effective reparameterization function for Dirichlet distribution [37], we thus construct a Laplace approximation of Dirichlet prior. The Dirichlet probability density function over the softmax variable $\boldsymbol{h}$ is given by

$$p(\boldsymbol{\theta}|\alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k} g(\mathbf{1}^T \boldsymbol{h}) \tag{3}$$

where $\boldsymbol{\theta} = \sigma(\boldsymbol{h})$ and $g$ is an arbitrary density function for integrability by constraining the redundant degree of freedom. The Laplace approximation also benefits with the property that the covariance matrix of the Dirichlet prior becomes diagonal for large topic number $K$ [9]. The Dirichlet prior $p(\boldsymbol{\theta}|\alpha)$ is approximated by a multivariate Gaussian with mean $\boldsymbol{\mu_1}$ and covariance matrix $\Sigma_1$:

$$\begin{aligned} \mu_{1k} &= \log \alpha_k - \frac{1}{K} \sum_i^K \log \alpha_i \\ \Sigma_{1kk} &= \frac{1}{K}(1 - \frac{2}{K}) + \frac{1}{K^2} \sum_i^K \frac{1}{\alpha_i} \end{aligned} \tag{4}$$

We can then approximate $p(\boldsymbol{\theta}|\alpha)$ in the simplex basis with

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{\mu_1}, \Sigma_1) = \mathcal{LN}(\boldsymbol{\theta}|\boldsymbol{\mu_1}, \Sigma_1) \tag{5}$$

where $\mathcal{LN}$ is a logistic normal distribution with parameters $\boldsymbol{\mu_1}, \Sigma_1$. By this way, we can apply RT to logistic normal and sample $\boldsymbol{\theta}$ to approximate Dirichlet prior. The detailed implementation of NTM is as follows:

**Encoder**: We first maps query words $\boldsymbol{q}_w$ and its entity annotations $\boldsymbol{q}_e$ to pretrained $L$-dimension embeddings $\boldsymbol{v}_w^q$ and $\boldsymbol{v}_e^q$, respectively. Then convolutional neural networks (CNN) is used to encode the meaning of a query into a fixed-length query embedding $\boldsymbol{h}_q$ by concatenating the representations of query words and entities:

$$\boldsymbol{h}_q = [CNN_q(\boldsymbol{v}_w^q), CNN_e(\boldsymbol{v}_e^q)] \tag{6}$$

Specifically, CNN is composed of a layer of convolutions and max-pooling as follows:

$$\begin{aligned} c_{m,i} &= ReLu(\boldsymbol{w}_m^T \boldsymbol{v}_{i:i+h-1} + b_m) \\ \hat{c}_m &= \max\{c_{m,1}, c_{m,2}, \ldots, c_{m,n-h+1}\} \\ CNN(\boldsymbol{v}) &= [\hat{c}_1 \oplus \hat{c}_2 \oplus \cdots \oplus \hat{c}_M], \end{aligned} \tag{7}$$

where $M$ is the number of convolution kernels and $n$ is the number of words/entities in the query. $\mathbf{w}_m$ and $b_m$ are the weights in the $m$-th convolution kernel, extracting a window of $h$ words/entities to produce a local feature. The final representation of query words/entities is represented as the concatenation of max-pooled outputs over all positions.

We define $q(\boldsymbol{\theta}|\boldsymbol{q}_w, \boldsymbol{q}_e)$ as the posterior logistic normal distribution with the posterior mean $\boldsymbol{\mu_0}$ and the covariance $\Sigma_0$, which are estimated by multi-layer neural networks $\mathcal{F}_\mu$ and $\mathcal{F}_\Sigma$, i.e.,

$$\boldsymbol{\mu_0} = \mathcal{F}_\mu(\boldsymbol{h}_q), \Sigma_0 = diag(\mathcal{F}_\Sigma(\boldsymbol{h}_q)) \tag{8}$$

where $diag$ converts a column vector to a diagonal matrix. Then we can apply RT to generate the $\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu_0}, \Sigma_0)$ by sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ and computing $\boldsymbol{\theta} = \sigma(\boldsymbol{\mu_0} + \Sigma_0^{1/2} \boldsymbol{\epsilon})$.

**Decoder**: The decoder takes the learned topic distribution $\boldsymbol{\theta}$ as input. Different from previous neural topic models [24, 37], we introduce a set of auxiliary implicit topic embeddings. The topic embeddings are used to represent topic information and match

with candidate document in the following topic enhanced matching framework. We denote $\mathcal{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K] \in \mathbb{R}^{d_t \times K}$ as the $d_t$-dimensional topic embeddings. The generative process of the query is derived as below:

$$\begin{aligned} p(\boldsymbol{q}_w|\boldsymbol{\beta}, \boldsymbol{\theta}) &= \sigma(\boldsymbol{W}_\beta \cdot \sigma(\mathcal{T}\boldsymbol{\theta})) \\ p(\boldsymbol{q}_e|\boldsymbol{\phi}, \boldsymbol{\theta}) &= \sigma(\boldsymbol{W}_\phi \cdot \sigma(\mathcal{T}\boldsymbol{\theta})) \end{aligned} \tag{9}$$

where $\boldsymbol{W}_\beta \in \mathbb{R}^{V \times d_t}$ and $\boldsymbol{W}_\phi \in \mathbb{R}^{E \times d_t}$ are the weight matrices to reconstruct query words and entities, respectively. $V$ and $E$ are the vocabulary size of words and entities. The topic-word probability matrix $\boldsymbol{\beta}$ and topic-entity probability matrix $\boldsymbol{\phi}$ can be regarded as a combination of two matrices, i.e., $\boldsymbol{\beta} = \boldsymbol{W}_\beta \mathcal{T}$ and $\boldsymbol{\phi} = \boldsymbol{W}_\phi \mathcal{T}$.

In our model, the implicit topic embeddings play a crucial role in topic modeling, which provide better representation power compared with topic distribution $\boldsymbol{\theta}$. They are learned during the generative process of VAE. To make topic embeddings independent with each other, we add an orthogonal constraint as follows:

$$\mathcal{L}_{orth} = ||\mathcal{T}^T\mathcal{T} - \boldsymbol{I}||_F^2 \tag{10}$$

where $\boldsymbol{I}$ is the identity matrix and $||\cdot||_F$ is the Frobenius norm.

The objective function of topic modeling is to maximize *the evidence lower bound* (ELBO), as derived as follows:

$$\begin{aligned} \mathcal{L}_{ELBO} &= KL(q(\boldsymbol{\theta}|\boldsymbol{q}_w, \boldsymbol{q}_e)||p(\boldsymbol{\theta}|\alpha)) + \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{q}_w, \boldsymbol{q}_e)}\left[\log p(\boldsymbol{q}_w, \boldsymbol{q}_e|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\phi})\right] \\ &= \sum_{q=1}^{|Q|}\Bigg[-(\frac{1}{2}\{tr(\Sigma_1^{-1}\Sigma_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\Sigma_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\}) \\ &\quad -K + \log\frac{|\Sigma_1^{-1}|}{|\Sigma_0^{-1}|}\Bigg] + \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\boldsymbol{q}_w^T\log(\hat{\boldsymbol{q}}_w) + \boldsymbol{q}_e^T\log(\hat{\boldsymbol{q}}_e)\right] \end{aligned} \tag{11}$$

where the first term is the KL divergence to match the posterior $q(\boldsymbol{\theta}|\boldsymbol{q}_w, \boldsymbol{q}_e)$ with the prior $\hat{p}(\boldsymbol{\theta}|\boldsymbol{\mu}_1, \Sigma_1)$ (the approximation of $p(\boldsymbol{\theta}|\alpha)$) and the second term is the reconstruction error for query words and entities generation. $|Q|$ is the number of query samples in the training data. $\boldsymbol{q}_w$ and $\hat{\boldsymbol{q}}_w$ are the bag-of-word representation of input query words and reconstructed query words. And it is similar to $\boldsymbol{q}_e$ and $\hat{\boldsymbol{q}}_e$. If there is no entity in a query, we will use <*unk*> as the entity in this query. The final objective function of NTM is thus derived as follows:

$$\mathcal{L}_{NTM} = \mathcal{L}_{ELBO} + \eta \cdot \mathcal{L}_{orth} \tag{12}$$

where $\eta$ is a parameter to balance the orthogonal constraint.

## 3.3 Topic Enhanced Knowledge-aware Framework

We then incorporate the topic embeddings from neural topic model into a knowledge-aware retrieval model. The whole model is called Topic Enhanced Knowledge-aware retrieval Model (TEKM), which combines semantic similarity, knowledge relevance and topic relatedness for learning to rank.

Given the words and entity annotations of query and document, our model contains three steps for relevance estimation. First, we build four different interaction matrices to measure semantic similarity, knowledge relevance and topic relatedness, which are query words to document words, query entities to document entities, query topics to document words and query topics to document entities. Then, we apply *kernel pooling* technique [43] as our interaction-based feature extractor to generate soft-match features. Third, the features are then concatenated to calculate the ranking score.

**Interaction matrix building**: We write $\boldsymbol{v}_{w_i}^q, \boldsymbol{v}_{w_j}^d \in \mathbb{R}^L$ to denote the word embedding of query $q$ and document $d$, respectively. We aim to build an interaction matrix $M$ where each element in $M$ is the embedding similarity between a query word and a document word:

$$M_{ij}^s = cos(\boldsymbol{v}_{w_i}^q, \boldsymbol{v}_{w_j}^d) \tag{13}$$

The matrix represents fine-grained word-level semantic similarity. Similarly, we construct interaction matrix to measure knowledge relevance and topic relatedness:

$$\begin{aligned} M_{ij}^e &= cos(\boldsymbol{v}_{e_i}^q, \boldsymbol{v}_{e_j}^d) \\ M_{ij}^{tw} &= cos(\hat{\boldsymbol{t}}_i, \boldsymbol{v}_{w_j}^d) \\ M_{ij}^{te} &= cos(\hat{\boldsymbol{t}}_i, \boldsymbol{v}_{e_j}^d) \end{aligned} \tag{14}$$

where $\boldsymbol{v}_{e_i}^q, \boldsymbol{v}_{e_j}^d$ denote the entity embedding of query $q$ and document $d$. $\hat{\boldsymbol{t}}_i = \theta_i \cdot \boldsymbol{t}_i$ is the $i$-th query topic representation in the weighted topic embedding matrix $\hat{\mathcal{T}} = \boldsymbol{\theta} \odot \mathcal{T}$ and $\odot$ is the element-wise multiplication. Specifically, $\hat{\mathcal{T}}$ represents the topic information weighted by the query's topic distribution over all given topics.

**Kernel pooling**: We then apply *kernel pooling* technique to map interaction matrix into soft-matched ranking features. In particular, our model uses $T$ Gaussian kernels to count the soft matches of interaction pairs at $T$ different strength levels. Each kernel summarizes the interaction features in matrix $M$ as soft similarity counts in the region defined by its mean $\mu_t$ and width $\sigma_t$, generating a $T$-dimensional feature vector $\phi(M) = \{K_1(M), \ldots, K_T(M)\}$:

$$\begin{aligned} K_t(M) &= \sum_i \log K_t(M_i) \\ K_t(M_i) &= \sum_j \exp(-\frac{(M_{ij} - \mu_k)^2}{2\sigma_t^2}) \end{aligned} \tag{15}$$

where matrix$M$ has different superscripts as shown in Equation 13 and 14. Prior studies have shown that such counting-based pooling methods can achieve better performance than score-based methods like mean-pooling or max-pooling [29].

We then apply kernel pooling to four interaction matrix in Equation 13 and 14, resulting $\phi(M^s), \phi(M^e), \phi(M^{tw})$ and $\phi(M^{te})$. The final ranking feature $\phi(\mathcal{M})$ is a concatenation ($\oplus$) of four extracted interaction features.

$$\phi(\mathcal{M}) = \phi(M^s) \oplus \phi(M^e) \oplus \phi(M^{tw}) \oplus \phi(M^{te}) \tag{16}$$

**Learning to rank**: This layer converts the final ranking feature $\phi(\mathcal{M})$ into a ranking score by a neural network:

$$f(q, d) = \tanh(\boldsymbol{w}^T\phi(\mathcal{M}) + \boldsymbol{b}) \tag{17}$$

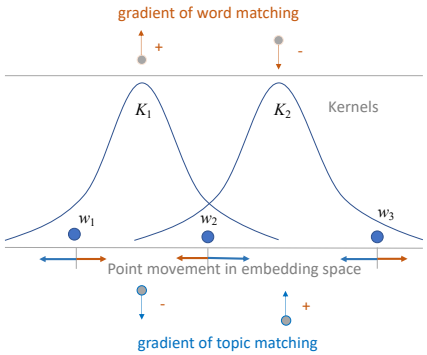



**Figure 3: Illustration of embedding learning in our framework, e.g., word embedding learning is guided by semantic matching signals and topic matching signals.**

We apply pairwise learning to rank loss to train our model: document $d^+$ is ranked higher than $d^-$ in terms of ground truth preferences, with respect to a query $q$;

$$\mathcal{L}_{rank} = \max(0, 1 - f(q, d^+) + f(q, d^-)) \tag{18}$$

The final learning objective of TEKM framework is to minimize the ranking loss and NTM loss:

$$\mathcal{L}_{TEKM} = \mathcal{L}_{rank} + \lambda \cdot \mathcal{L}_{NTM} \tag{19}$$

where $\lambda$ is a parameter to balance the two losses of learning to rank and NTM.

## 3.4 Discussion

TEKM estimates relevance by modeling three relevance aspects: semantic relevance, knowledge relevance and topical relatedness. The parameters of the whole model are learned to better estimate the final relevance. In particular, the embedding learning is guided by two relevance signals, which is different from traditional kernel pooling based ranking models. For example, word embeddings are learned by the kernels in semantic matching interaction matrix $M^s$ and topic-word interaction matrix $M^{tw}$. As shown in Figure 3, the learning of word embeddings[1] is guided by two matching signals, i.e., semantic similarity and topic relatedness. For traditional word-level semantic learning process [43], a kernel moves two word embeddings such that their similarity is closer (or away) to the kernel mean $\mu$. However, the matching of word embedding can only capture semantic similarity. In our framework, word embeddings are also interacted with topic embeddings and updated by their similarity (or distance) with the specific topic embedding. The topic embedding[2] can be regarded as a cluster center in the word embedding space. The learning process makes the word embeddings clustered to a specific topic in terms of their similarity to a specific topic embedding. It is similar to entity embedding used in our model, which is guided simultaneously by knowledge relevance and topical relatedness. Compared with previous semantic similarity based only model [43] or knowledge-aware retrieval model [22], our model learns the words and entities cluster points as the topic embeddings. They are further exploited to match topical relevant words and entities in the document, providing topical relatedness to better estimate relevance.

[1]Different from the illustration in [43], we represent two words around the kernel and words around the smaller kernel mean are closer in the word embedding space.
[2]Since topic embeddings are tuned to fit word and entity together during training, our experiment finds that using shared topic embeddings for both word and entity can achieve better ranking performance compared with the separated topic embeddings.

**Table 1: Statistics of our experimental data Tiangong-ST**

| | Train | Valid | Test |
|---|---|---|---|
| #queries | 344,942 | 4,888 | 2,000 |
| #sessions | 143,155 | 2,000 | 2,000 |
| #avg. click[3] per query | 3.29 | 3.36 | 3.52 |
| #avg. doc per query | 9.60 | 9.58 | 9.59 |

# 4 EXPERIMENTAL SETUP

## 4.1 Dataset

We conduct our experiments on a large-scale, publicly available query log from a Chinese commercial search engine, Sogou.com, namely Tiangong-ST [4] [5]. Table 1 shows the statistics of our dataset. Tiangong-ST provides web search session data extracted from an 18-day search log. It contains weak relevance labels (i.e., click relevance labels [43]) derived by six different click models for all query-document pairs and human relevance labels for documents in the last query of 2,000 sampled sessions. We use the 2,000 last queries as our test set for an ad-hoc retrieval task. To avoid enormous entity number in the document, we exploit document titles in both training and testing instead of the full document content. Prior studies [6, 43] have shown that weak relevance labels derived from click models can be used to train and evaluate retrieval models. Since the Partially Sequential Click Model (PSCM) [40] achieves the best relevance estimation performance among the six click model alternatives, we employ click labels from the PSCM for training and validation. Following the experimental setups in previous works [22, 43], we utilize three different relevance labels to evaluate our model on the test set. In the Test-SAME setting, we uses click relevance labels from the same PSCM to evaluate our model. In the Test-DIFF setting, we use the Dynamic Bayesian Network click model (DBM) [8] as the relevance labels for evaluation. In HUMAN-label setting, we use the provided five-graded human annotated relevance labels to evaluate ranking performance.

For the entity annotations, we utilize XLore [41] as our knowledge graph foundation. XLore is an English-Chinese bilingual knowledge graph built from English Wikipedia, Chinese Wikipedia, Baidu Baike and Hudong Baike. It contains 16,284,901 entities, 2,466,956 concepts and 446,236 relations. The query and document entities are annotated by CMNS [12], the commonness (popularity) based entity linker, which follows the settings of previous work [22]. The distribution of the entity number is shown in Figure 4, where most of query and document have at least one entity. The average entity number of query and document is 3.44 and 4.92, respectively.

## 4.2 Experimental Settings

*4.2.1 Baselines.* To compare the effectiveness of three relevance dimensions (i.e., semantic similarity, knowledge relevance and topical relatedness), we exploit four different types of retrieval models as our baselines: Topic based retrieval models (T), semantic based

[3]Pseudo click drawn from PSCM labels, where documents with top 29% click probability (grade 1 or higher) are considered being clicked.
[4]http://www.thuir.cn/tiangong-st/.

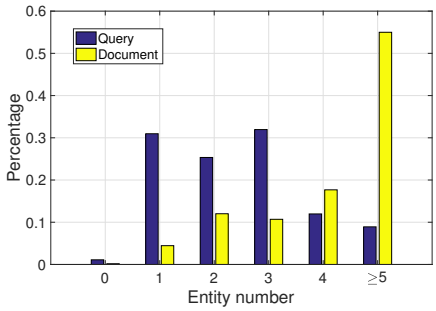



**Figure 4: The distribution of query and document with different entity numbers.**

retrieval models (S), knowledge-aware retrieval models (S+K) and topic enhanced knowledge-aware retrieval models (S+K+T), where S, K, T represent semantic similarity, knowledge relevance and topical relatedness, respectively.

- **BM25** (S): A popular probabilistic bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document [31].
- **Matchpyramid** [30] (S): A neural retrieval model that first builds a word-level semantic interaction matrix and then applies CNNs to aggregate it into the final relevance score. We use a one-layer CNN with 64 $(1 \times 3)$ kernels and a $(2 \times 2)$ pooling size.
- **KNRM** [43] (S): A neural retrieval model that uses a kernel pooling to model multi-level semantic similarity signals. We use 11 kernels as the default setting in the original paper (10 soft-matching and 1 exact-matching kernels).
- **ARCII** [15] (S): A neural retrieval model that maps the word embeddings of query and document to an aggregated embedding by a CNN. we use a two-layer CNN, where the size of kernels and pooling in both layers are set to $(3 \times 3)$ and $(2 \times 2)$. There are 16/32 kernels in two layers.
- **ARCI** [15] (S): ARCI is a representation-based model which encodes text information by CNNs. We use a three-layer CNN where the filter windows sizes are 1 to 3 and there are 64 feature maps for each filter.
- **DSSM** [16] (S): DSSM is also a representation-based model. It consists of a word hashing layer, two non-linear hidden layers, and an output layer. We use a three-layer DNN as in the original paper of DSSM; the hidden number of each layer is set to 50.
- **KESM** [44] (S+K): KESM is proposed to model the salience of query entities in candidate documents. It uses kernel pooling to model the interactions of query entities with the entities and words in the document. We set the dimension of word and entity embedding as 50 and remains other parameter settings as the original paper.
- **EDRM** [22] (S+K): EDRM exploits four interactions matrices to model word-entity duet relationships, which considers both semantic similarity and knowledge relevance. We set the dimension of word and entity embedding as 50 and remains other parameter settings as the original paper.
- **LBDM** [42] (T): LDA-Based Document Model (LBDM) was firstly proposed to combine traditional language model like query likelihood and BM25. It calculates relevance score as illustrated in Equation 1. Jian et al. [17] improves LBDM by summing all the term likelihoods $p(w|D)$ in Equation 1 and $P_{QL}$ can be any other retrieval models. The inference function becomes:

$$rel(q,d) = (1-\lambda) \cdot RM(q,d) + \lambda \cdot LDA(q,d). \tag{20}$$

where $RM$ represents the score from an external Retrieval Model and $LDA(q,d) = \sum_w^{|q|} \sum_z P(w|z)P(z|d)$ is the topic model based relevance score. $\lambda$ is tuned based on the ranking performance in validation set. We can thus incorporate other retrieval models by this function. To avoid redundancy when applying different $RM(q,d)$, we refer **LBDM** to the topic model based relevance score $LDA(q,d)$ only (i.e., $\lambda = 1$) and combine it with the best performing baseline retrieval models (i.e., KESM and EDRM) as our baselines.
- **LDA-DSSM** (T): To further compare the effectiveness of individual topic information, we directly use the implicit topic vectors from LDA to represent query and document. The vectors are then inputted to DSSM for relevance estimation. The topic number is set as 50.
- **LDA-KESM** and **LDA-EDRM** (S+K+T): As discussed in Equation 20, we combine the topic model based relevance score $LDA(q,d)$ with two best performing baselines instead of using all baselines. $RM(q,d)$ is replaced by the scores from KESM and EDRM. $\lambda$ is tuned based on the ranking performance in validation set, i.e., 0.2 and 0.4, respectively.

*4.2.2 Parameter settings.* We implement our models using Pytorch. The parameters are optimized by Adam, with a batch size of 32 and a learning rate of 0.001. The dimension of the word embeddings, entity embedding and topic embeddings in NTM are all 50. Word embeddings are pretrained on a Chinese Wikipedia dataset[5] by word2vec while entity embeddings are pretrained based on node2vec [10][6]. For CNN layer in NTM, the filter windows sizes are 1 to 3 and there are 50 feature maps for each filter. Topic embedding is initialized by orthogonal initialization. For kernel pooling, we use 11 kernels which contain an exact match kernel ($\mu = 1, \sigma = 0.001$) and 10 soft match kernels ($\mu = [0.9, 0.7, \cdots, -0.9]$, $\sigma = 0.1$). The hyper-parameter $\eta$ and $\lambda$ are selected from [0.001, 0.01, 0.1,1,10]. The number of topics is selected from [32, 64, 128, 256, 512]. Early stopping with a patience of 5 epochs is adopted during the training process. We use NDCG (Normalized Discounted Cumulative Gain) as evaluation metric. The source code is publicly available[7].

# 5 EVALUATION RESULTS

## 5.1 Overall Ranking Performance

We first compare our model TEKM with various comparison models. The overall ranking performance under three different evaluation labels are shown in Table 2 and 3.

We first observe that most of retrieval models perform similarly among three evaluation labels and the performance on PSCM is more like that of human labels. It shows that retrieval models trained on click labels are also effective for human evaluation.

For **Topic based retrieval model**, we find that LDBM and LDA-DSSM perform worst compared with other retrieval models on three

[5] http://download.wikipedia.com/zhwiki
[6] We do not use the popular TransE [4] to pretrain entity embedding because the number of relation types in XLore is small.
[7] https://github.com/lixsh6/TEKM-ranker.

**Table 2: Overall ranking performance of our model and other baselines. Results of best performing baselines are underlined. 1, 2 indicates a significant improvement over KESM and EDRM, respectively. † indicate statistically significant improvements over all baselines. (p-value ≤ 0.05)**

| | PSCM(SAME) | | | | DBN(DIFF) | | | |
|---|---|---|---|---|---|---|---|---|
| | **NDCG@1** | **NDCG@3** | **NDCG@5** | **NDCG@10** | **NDCG@1** | **NDCG@3** | **NDCG@5** | **NDCG@10** |
| **Topic based retrieval model (T)** | | | | | | | | |
| LBDM | 0.1530 | 0.2064 | 0.3419 | 0.3762 | 0.1374 | 0.1634 | 0.2953 | 0.3294 |
| LDA-DSSM | 0.2191 | 0.2403 | 0.3971 | 0.5023 | 0.1582 | 0.2341 | 0.3223 | 0.4021 |
| **Semantic based retrieval models (S)** | | | | | | | | |
| BM25 | 0.2403 | 0.3887 | 0.4743 | 0.6363 | 0.2144 | 0.3688 | 0.4440 | 0.6039 |
| MatchPyramid | 0.3254 | 0.4108 | 0.4936 | 0.6558 | 0.2707 | 0.3579 | 0.4512 | 0.6165 |
| KNRM | 0.5224 | 0.5978 | 0.6622 | 0.7642 | 0.4074 | 0.5279 | 0.5901 | 0.6986 |
| ARCII | 0.5657 | 0.6117 | 0.6866 | 0.7854 | 0.4304 | 0.5175 | 0.5888 | 0.7084 |
| ARCI | 0.5834 | 0.6475 | 0.7012 | 0.7946 | 0.4512 | 0.5493 | 0.6059 | 0.7204 |
| DSSM | 0.5859 | 0.6646 | 0.7143 | 0.8043 | 0.4497 | 0.5583 | 0.6134 | 0.7263 |
| **Knowledge-aware retrieval model (S+K)** | | | | | | | | |
| KESM | 0.6690 | 0.7674 | 0.7447 | 0.8459 | 0.4849 | 0.5831 | 0.6453 | 0.7403 |
| EDRM | 0.7233 | 0.7812 | 0.8106 | 0.8645 | 0.5053 | 0.5990 | 0.6702 | 0.7490 |
| **Topic enhanced knowledge-aware ranking model (S+K+T)** | | | | | | | | |
| LDA-KESM | 0.6783[1] | 0.7712 | 0.7489 | 0.8473 | 0.4612 | 0.5703 | 0.6204 | 0.7367 |
| LDA-EDRM | 0.7211 | 0.7885 | 0.8203[2] | 0.8685 | 0.4942 | 0.5790 | 0.6531 | 0.7354 |
| **TEKM** | **0.7328**† | **0.8042**† | **0.8377**† | **0.8818**† | **0.5431**† | **0.6366** † | **0.7015**† | **0.7811** † |

**Table 3: Over ranking performance on Human annotated labels. Significance marks 1, 2, † are the same as Table 2.**

| | **NDCG@1** | **NDCG@3** | **NDCG@5** | **NDCG@10** |
|---|---|---|---|---|
| **Topic based retrieval model (T)** | | | | |
| LBDM | 0.2205 | 0.2449 | 0.4402 | 0.5585 |
| LDA-DSSM | 0.3748 | 0.3842 | 0.5402 | 0.6092 |
| **Semantic based retrieval models (S)** | | | | |
| BM25 | 0.4528 | 0.5533 | 0.6244 | 0.7856 |
| MatchPyramid | 0.4807 | 0.5565 | 0.6135 | 0.7849 |
| KNRM | 0.5560 | 0.6080 | 0.6707 | 0.8144 |
| ARCII | 0.5822 | 0.6363 | 0.6889 | 0.8285 |
| ARCI | 0.6045 | 0.6628 | 0.7043 | 0.8256 |
| DSSM | 0.6141 | 0.6550 | 0.7016 | 0.8253 |
| **Knowledge-aware retrieval model (S+K)** | | | | |
| KESM | 0.6164 | 0.6547 | 0.6989 | 0.8345 |
| EDRM | 0.6314 | 0.6655 | 0.7135 | 0.8389 |
| **Topic enhanced knowledge-aware retrieval model (S+K+T)** | | | | |
| LDA-KESM | 0.6213 | 0.6604 | 0.7017 | 0.8401 |
| LDA-EDRM | 0.6351 | 0.6714 | 0.7221[2] | 0.8415 |
| **TEKM** | **0.6549**† | **0.7003**† | **0.7363**† | **0.8522**† |

different evaluation labels. This illustrates that using individual topic information to represent query and document will lose much information in the original text and hurt ranking performance.

For **Semantic based retrieval models**, we first observe that BM25 performs relatively worse compared with other models. This indicates that merely considering exact word matching is not enough to model the relevance between query and document. Instead, neural retrieval models, which models semantic similarity based on distributed word embeddings, can achieve substantial improvements over BM25. It shows that semantic similarity is a key part to estimate relevance for neural retrieval models. Compared with all semantic based retrieval models, our model TEKM achieves significantly better ranking performance, which indicates that considering more relevance dimensions are helpful to construct better retrieval models.

For **Knowledge based retrieval models**, both KESM and EDRM achieve better performance than semantic based retrieval models significantly over three evaluation labels. It illustrates that knowledge relevance is also an important part for relevance estimation besides semantic similarity. The main difference between KESM and EDRM is that KESM only exploits query entities and ignores query words to match the document's words and entities. This results in the relatively poor performance of KESM. Our model TEKM outperforms KESM and EDRM with a significant level over three evaluation labels. This confirms that topic relatedness is useful for the retrieval task and it can be well exploited in our framework.

For **Topic enhanced knowledge-aware retrieval models**, we find that the topic enhanced framework in Equation 20 does not improve the best performing baselines KESM and EDRM significantly over most evaluation metrics. The results on DBN labels are even worse than the original retrieval models. This shows that topic information cannot be simply incorporated into neural networks. However, our model TEKM learns topic embedding to represent each topic from both neural topic model and the following matching framework in an end-to-end manner. Therefore, the topic

**Table 4: Ablation study of different ways to construct topic embeddings. Relative performances are compared in percentages.**

| | PSCM(SAME) | | DBN(DIFF) | | HUMAN | |
|---|---|---|---|---|---|---|
| | **NDCG@3** | **NDCG@10** | **NDCG@3** | **NDCG@10** | **NDCG@3** | **NDCG@10** |
| TEKM (w/o topic) | 0.7713 | 0.8628 | 0.6101 | 0.7567 | 0.6555 | 0.8329 |
| TEKM (word only) | 0.7889 (+2.28%) | 0.8795 (+1.95%) | 0.6218 (+1.92%) | 0.7705 (+1.82%) | 0.6981 (+6.50%) | 0.8504 (+2.10%) |
| TEKM (entity only) | 0.7722 (+0.12%) | 0.8694 (+0.76%) | 0.6172 (+1.16%) | 0.7674 (+1.41%) | 0.6746 (+2.91%) | 0.8485 (+1.87%) |
| TEKM | **0.8042** (+4.27%) | **0.8818** (+2.20%) | **0.6366** (+4.34%) | **0.7811** (+3.22%) | **0.7003** (+6.83%) | **0.8522** (+2.32%) |

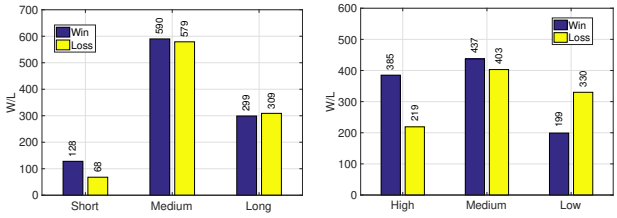
embeddings have better topic representation power and can be well exploited in neural retrieval model. In summary, our model outperforms other baselines significantly over three evaluation metrics. It demonstrates that topic relatedness is useful for retrieval task and our model can measure three relevance dimensions well for constructing better neural retrieval models.

## 5.2 Ablation Study on Topic Embeddings

This experiment studies the effectiveness of different inputs to build topic embeddings. Recall that neural topic model takes both query words and entities as inputs and reconstructs both of them based on the topic embeddings. Therefore, we test our model by using different inputs to the neural topic model, yielding TEKM without topic embeddings (w/o topic), TEKM with only word input (word only) and TEKM with only entity input (entity only). The ranking performance over three different evaluation labels is shown in Table 4.

Using both query words and entities for neural topic model aims to improve the quality of generated topic embeddings. TEKM without topic embedding only considers semantic similarity and knowledge relevance like other knowledge-aware retrieval model. Its difference with EDRM is that EDRM uses additional word-entity interaction matrices. TEKM (w/o topic) performs similarly with EDRM, which demonstrates that word-word and entity-entity interaction matrices are the key part to model semantic similarity and knowledge relevance. Furthermore, we observe that in general our framework built on the generated topic embeddings achieves better performance than the original model without topics. It shows that both inputs (words and entities) are helpful for topic modeling. In particular, our model generating on the single word inputs performs substantially better than that of the single entity inputs. The improvement over the single entity inputs is also limited compared with the original model. This is due to two reasons:1) using entity information to represent the whole query will lose a part of information out of the extracted entities and there are also a few queries without any entities (See Figure 4). 2) the entity extraction method used in our study cannot guarantee the quality of the extracted entities, where wrong or duplicated entities may be extracted. Both of these two reasons cause the information loss in topic modeling and effects the topic quality. Moreover, this also indicates that word information is crucial to topic modeling since the learned topic embeddings from individual word information improves ranking performance substantially.

By combining both of query words and entities, TEKM outperforms other variants with single input source. It illustrates that our model can make full of different sources and learns better topic embeddings for further relevance estimation.




**(a) Query length** **(b) Query frequency**

**Figure 5: Ranking performance comparison between TEKM and EDRM on the human annotated labels. Y-axis represents the Win/Loss number over different groups.**

## 5.3 Performance on Different Scenarios

This experiment analyzes the influence of topical relatedness in two different scenarios: different query lengths and query frequencies. We compare the performance of our model TEKM with the best performing baseline EDRM and count Win/Loss number in Figure 5.

Prior studies [22] have shown that query length is an important impact on ranking performance. We split the testing queries into three groups in terms of their lengths: short queries ($\leq$ 2 words), medium queries (3-4 words) and long queries ($\geq$ 5 words). It is observed that TEKM has more win cases and achieves the larger improvements on short and medium queries. According to the previous study [36], short queries tend to be broad and have ambiguous search intent. By adding additional relevance dimension (topical relatedness), our model can perform better when limited information is available from the original query text.

We calculate query frequency by averaging the term frequency in the training set and category them into three groups: the 30% queries with the lowest frequency (Low), the 30% queries with the highest frequency (High) and others (Medium). We observe that queries with high and medium frequency performs better compared with EDRM. Previous research [14] found that topic model is unable to learn a good topic representation on the sparse text since topic model is learned based on context information. Therefore, this reason causes our model use a relatively noisy topic embedding, which influences the performance of relevance estimation.

Note that short queries do not mean that the query frequency is low. There may be popular words in the short queries and thus they have good ranking performance. As explained in previous study [14], topic modeling approaches are very useful for short text and it does not mean that the model itself should be trained on short text. Instead, a model trained on aggregated longer text can yield better performance.

**Table 5: Examples from queries that TEKM improves or hurts compared with EDRM. The top-ranked documents by two models and the topic words from the topic with maximal probability are shown. Document IDs from the original data are in parenthesis.**

| Query | Topic words | TEKM Preferred Document | EDRM Preferred Document |
|---|---|---|---|
| **Cases that TEKM improves the ranking performance** | | | |
| Driver Genius | Official download computer version TXT software ... | (d72422) "Driver genius 2017 official website to download - PChome download center" | (d184681) "Driver genius - Sogou Wiki" |
| Bank rate | Fund net value Oriental ... stock makes | (d38606) "Bank Financial Port - Bank rate about deposits, loan and others" | (d162219) "Gaps between Bank rates in different banks increases " |
| National education platform | ... educational backstage entrance login education resources | (d66699) "National basic education resource website" | (d82766) "How to use National education public service" |
| **Cases that TEKM hurts the ranking performance** | | | |
| Insurance and bank deposit | Mean what effect + impact useful ... | (d93837) "Old people deposit money to banks" | (d93835) "Difference between insurance and bank deposit - sunflower insurance " |
| How to connect laptop with iphone4s | Official download computer version TXT software ... | (d401713) "IOS7 firmware download - iPhone5s/iphone4s" | (d137596) "How to connect laptop with iphone4s - BAIDU Experience" |
| microphone sound is low | Official download computer version TXT software ... | (d286919) "LENOVO official website, product instruction, tutorial" | (d286915) "Win7 - microphone sound is low, solution - BAIDU Experience" |

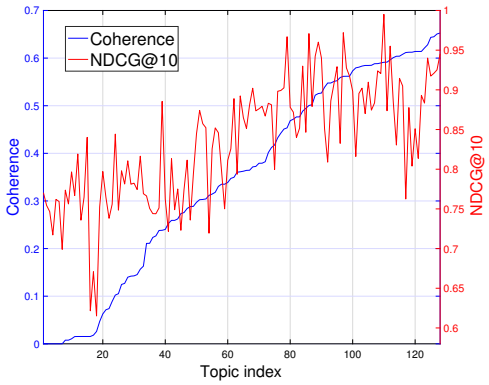



**Figure 6: Ranking performance over different topic coherences on the human annotated labels. Topic indices are sorted by the topic coherence.**

**Table 6: Ranking performance over different topic numbers on three evaluation labels.**

| | PSCM | | DBN | | HUMAN | |
|---|---|---|---|---|---|---|
| NDCG | @3 | @10 | @3 | @10 | @3 | @10 |
| 16 | 0.7866 | 0.8728 | 0.6186 | 0.7712 | 0.6700 | 0.8442 |
| 32 | 0.7887 | 0.8756 | 0.6233 | 0.7697 | 0.6775 | 0.8423 |
| 64 | 0.8012 | 0.8782 | 0.6278 | 0.7768 | 0.6801 | 0.8459 |
| 128 | 0.8042 | 0.8818 | 0.6366 | 0.7811 | **0.7003** | **0.8522** |
| 256 | 0.8038 | 0.8810 | 0.6371 | 0.7804 | 0.6966 | 0.8474 |
| 512 | **0.8074** | **0.8834** | **0.6384** | **0.7813** | 0.6941 | 0.8502 |

## 5.4 Impact of Topic Coherence

Topic embedding is a crucial part in our model which is used to measure topical relatedness for relevance estimation. Thus, it is also interesting to analyze if the quality of topics will influence the final ranking performance. To measure the topic quality, we use *topic coherence* as our metric. Topic coherence is a quantitative measure of the interpretability of a topic [26]. It is the average point-wise mutual information of the words from the same topic:

$$Coherence(k) = \sum_{i,j}^{20} \frac{2}{N^2} \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \tag{21}$$

where $w_i, w_j$ are the top 20 most likely words in the topic $k$. $P(w_i, w_j)$ is the probability of words $w_i$ and $w_j$ co-occurring in the same text (i.e., the query in our study). $P(w_i)$ is the marginal probability of word $w_i$. The intuitive idea behind topic coherence is that a coherent topic will display words that tend to occur in the same query. Since the quality of the annotated entities is not always perfect due to the extraction method (as discussed in Section 5.2)), we only calculate the coherence score of words. We group the ranking performance of each testing query in terms of their coherence of the most likely topic. The average ranking performance of each groups over the sorted coherences on the human annotated labels is shown in Figure 6.

We observe that generally, the ranking performance of our model TEKM has positive correlation with the coherence score. The pearson correlation coefficient is 0.69 with a significant level ($p < 0.01$). It shows that our topic enhanced framework relies on the quality of the learned topics to some extent. The noisy topic embeddings cannot be a good cluster point among the word embeddings, as discussed in Section 3.4. This reason causes our model learn a noisy topical relatedness and thus influences the ranking performance.

## 5.5 Impacts of topic number

This experiment studies the ranking performance when using different topic numbers. Due to the space limitation, we only report NDCG@3, 10 and other metrics are qualitatively similar. We tune the topic number in the range of {16, 32, 64, 128, 256, 512}. The results are shown in Table 6.

We can observe that on three evaluation labels, our model TEKM performs relatively worse when the topic number is small. Recall that the topic number can be regarded as the number of clusters

in embedding space, as discussed in Section 3.4. When using small topic number, it is hard to have meaningful clusters (i.e., high topic coherence). Therefore, our model cannot perform well over small topic number. On the other hand, as topic number increases, our model performs better on PSCM and DBN labels, but the performance on human annotated labels drops. This is probably due to the gap between click labels and real manual labels [20]. The training on click labels may be over-fitted on other testing labels.

### 5.6 Case Study

With some special cases we can better understand how topical relatedness is exploited in our model. In Table 5, we compare both good and bad cases compared with EDRM, which does not consider topical relatedness. The top-ranked documents by two models with respect to different queries.

The improvements from TEKM are mainly from its ability to estimate suitable topical relatedness. As discussed in Section 5.3, we find our model performs better than EDRM over short and medium queries. When the search intent of the issued query is vague or broad, our model improves the ranking performance by additional relevance dimension modeling. For example in the improved cases in Table 5, "*Driver Genius*" is a popular application software. Our model produces a topic about looking for softwares and ranks the Web page with application direct download at the top position. Besides semantic matching signal (or exact matching signal in this case), our model provides additional topical relatedness (*download*) that is more suitable to satisfy user search intent.

The cases that TEKM hurts are mostly due to two reasons: 1) Wrong topic clustering or low-quality topic interpretability (i.e., topic coherence). As the first case that TEKM hurts in Table 5, the query drops in a topic that is hardly to interpret (with topic coherence 0.087). Our model thus cannot provide a suitable topical relatedness and leads to wrong signals for relevance estimation. This noise hurts the ranking performance of our model. 2) We observe that when a query has clear search intent, the topical relatedness does not help to improve ranking performance. And it is worth noting that most of these cases happened on long queries. For example, in the last two cases in Table 5, their search intent is already clear. Although our model provides the correct topics with good quality, the topical relatedness is not so helpful. This finding is similar to the discussion about the query length in Section 5.3. It suggests that topical relatedness is more suitable for the queries without clear search intent.

## 6 CONCLUSION

This paper proposes a Topic enhanced knowledge-aware retrieval model (TEKM) which explicitly models three dimensions of relevance. Specifically, we introduce semantic similarity, knowledge relevance and topical relatedness into the process of relevance estimation. TEKM employs a neural topic model to generate topic embeddings, which are further exploited to soft match with word and entity embeddings. To the best of our knowledge, this is the first attempt to integrate topic information into neural retrieval models. Extensive experiments demonstrate the effectiveness of the proposed framework and its advantages in different scenarios. Moreover, since we exploit neural topic model to generate topic embeddings, our model also inherits similar characteristics with traditional topic models. For example, our model tends to perform better when the topic quality (i.e., coherence) is good and the predefined topic number is suitable for the dataset. Finally, we conduct case study to understand how topic information contributes to the ranking improvements. We illustrate the importance to model topical relatedness when the search intent is vague or broad and its dispensability in two aspects. Our study systematically analyzes the effectiveness of topic information in different ranking scenarios, providing a solid understanding of how to effectively utilize topic information for neural retrieval models.

In the future, we plan to extend our work to contextualized transformer-based ranking model like BERT [7]. We also plan to design self-adapted strategies to weight different relevance dimensions. For example, when the topic coherence is low, the weight of estimated topical relatedness should be lower than other dimensions. We believe that estimating relevance from different dimensions is necessary for better inferring user search intent and building better web search systems.

## ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Program of China (2018YFC0831700), Natural Science Foundation of China (Grant No. 61732008, 61532011, 61902209), Beijing Academy of Artificial Intelligence (BAAI) and Tsinghua University Guoqiang Research Institute. This project is also supported by Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098) and China Postdoctoral Science Foundation (2020M670339). Dr. Weizhi Ma has been supported by Shuimu Tsinghua Scholar Program.

## REFERENCES

[1] Haoli Bai, Zhuangbin Chen, Michael R Lyu, Irwin King, and Zenglin Xu. 2018. Neural Relational Topic Models for Scientific Article Analysis. (2018), 27–36.
[2] Nicholas J Belkin. 2016. People, Interacting with Information1. In *ACM SIGIR Forum*, Vol. 49. ACM New York, NY, USA, 13–27.
[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[4] Nicolas Usunier Alberto Garcia-Duran Jason Weston Bordes, Antoine and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
[5] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. TianGong-ST: A New Dataset with Large-scale Refined Real-world Web Search Sessions. In *Proceedings of the 28th ACM International on Conference on Information and Knowledge Management*. ACM, 2485–2488.
[6] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. San Rafael: Morgan and Claypool.
[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[8] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 331–338.
[9] T Graepel, P Hennig, R Herbrich, and D Stern. 2012. Kernel Topic Models. In *Artificial Intelligence and Statistics*. 511–519.
[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*.
[11] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2019. A Deep Look into Neural Ranking Models for Information Retrieval. *arXiv preprint arXiv:1903.06902* (2019).
[12] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2017. *Entity Linking in Queries: Efficiency vs. Effectiveness*. Springer, Cham.
[13] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. 51, 2 (1999), 50–57.

[14] Liangjie Hong and Brian D. Davison. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*. 80–88.
[15] 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*. 2042–2050.
[16] 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*. ACM, 2333–2338.
[17] Fanghong Jian, Jimmy Xiangji Huang, Jiashu Zhao, Tingting He, and Po Hu. 2016. A simple enhancement for ad-hoc information retrieval via topic modelling. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 733–736.
[18] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[19] Xiangsheng Li, Yiqun Liu, Jiaxin Mao, Zexue He, Min Zhang, and Shaoping Ma. 2018. Understanding Reading Attention Distribution during Relevance Judgement. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 733–742.
[20] Xiangsheng Li, Jiaxin Mao, Chao Wang, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. Teach Machine How to Read: Reading Behavior Inspired Relevance Estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 795–804.
[21] Xiangsheng Li, Yanghui Rao, Haoran Xie, Raymond Y K Lau, Jian Yin, and Fu Lee Wang. 2017. Bootstrapping Social Emotion Classification with Semantically Rich Hybrid Neural Networks. *IEEE Transactions on Affective Computing* 8, 4 (2017), 428–442.
[22] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. *arXiv preprint arXiv:1805.07591* (2018).
[23] Marcelo Mendoza, Pablo Ormeno, and Carlos Valle. 2018. Ad-hoc Information Retrieval based on Boosted Latent Dirichlet Allocated Topics. In *2018 37th International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 1–7.
[24] Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. *arXiv preprint arXiv:1706.00359* (2017).
[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[26] Hanna Wallach Edmund Talley-Miriam Leenders Mimno, David and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 International Joint Conference on Natural Language Processing*. 262–272.
[27] Bhaskar Mitra and Nick Craswell. 2018. An Introduction to Neural Information Retrieval. *Foundations and Trends® in Information Retrieval* 13, 1 (2018), 1–126.
[28] Jiaul H Paik. 2013. A novel TF-IDF weighting scheme for effective ranking. (2013), 343–352.
[29] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2017. A Deep Investigation of Deep IR Models. *arXiv preprint arXiv:1707.07700* (2017).
[30] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference*.
[31] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*.
[32] Tim Salimans and David A. Knowles. 2013. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. *Bayesian Analysis* (2013).
[33] Tefko Saracevic. 2006. Relevance: A review of the literature and a framework for thinking on the notion in information science. Part II. *Advances in librarianship* 30 (2006), 03.
[34] Tefko Saracevic. 2007. Relevance: A review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. *Journal of the American Society for information Science and Technology* 58, 13 (2007), 2126–2144.
[35] Tefko Saracevic. 2016. The Notion of Relevance in Information Science: Everybody knows what relevance is. But, what is it really? *Synthesis Lectures on Information Concepts, Retrieval, and Services* 8, 3 (2016), i–109.
[36] Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. 2009. Identification of ambiguous queries in web search. *Information Processing & Management* 45, 2 (2009), 216–229.
[37] Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488* (2017).
[38] Mizzaro Stefano. 1998. How many relevances in information retrieval? *Interacting with Computers* 10, 3 (1998), 303–320.
[39] B. C. Vickery. 1959. Subject analysis for information retrieval. In *Proceedings of the International Conference on Scientific Information*. Washington, DC: National Academy of Sciences, 855–866.
[40] Chao Wang, Yiqun Liu, Meng Wang, Ke Zhou, Jian-yun Nie, and Shaoping Ma. 2015. Incorporating non-sequential behavior into click models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 283–292.
[41] Zhigang Wang, Juanzi Li, Zhichun Wang, Shuangjie Li, Mingyang Li, Dongsheng Zhang, Yao Shi, Yongbin Liu, Peng Zhang, and Jie Tang. 2013. XLore: A Large-Scale English-Chinese Bilingual Knowledge Graph. In *Proceedings of the 12th International Semantic Web Conference*. 121–124.
[42] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. (2006), 178–185.
[43] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conferenc*. ACM, 55–64.
[44] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tieyan Liu. 2018. Towards Better Text Understanding and Retrieval through Kernel Entity Salience Modeling. (2018), 575–584.
[45] Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems* 22, 2 (2004), 179–214.