Information Processing and Management 59 (2022) 102767

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

# Webpage retrieval based on query by example for think tank construction

Qian Geng [a,b], Ziang Chuai [b], Jian Jin [b,*]

[a] Center for Governance Studies, Beijing Normal University at Zhuhai, Zhuhai, People's Republic of China
[b] Department of Information Management, School of Government, Beijing Normal University, Beijing, People's Republic of China

ARTICLE INFO

*Keywords:*
Feature bootstrapping
Pre-trained neural networks
Query by example
Textual features
Visual features
Webpage retrieval

ABSTRACT

Think tanks have been proved helpful for decision-making in various communities. However, collecting information manually for think tank construction implies too much time and labor cost as well as inevitable subjectivity. A probable solution is to retrieve webpages of renowned experts and institutes similar to a given example, denoted as query by webpage (QBW). Considering users' searching behaviors, a novel QBW model based on webpages' visual and textual features is proposed. Specifically, a visual feature extraction module based on pre-trained neural networks and a heuristic pooling scheme is proposed, which bridges the gap that existing extractors fail to extract snapshots' high-level features and are sensitive to the noise effect brought by images. Moreover, a textual feature extraction module is proposed to represent textual content in both term and topic grains, while most existing extractors merely focus on the term grain. In addition, a series of similarity metrics are proposed, including a textual similarity metric based on feature bootstrapping to improve model's robustness and an adaptive weighting scheme to balance the effect of different types of features. The proposed QBW model is evaluated on expert and institute introduction retrieval tasks in academic and medical scenarios, in which the average value of MAP has been improved by 10% compared to existing baselines. Practically, useful insights can be derived from this study for various applications involved with webpage retrieval besides think tank construction.

## 1. Introduction

Think tanks, which can be understood as a group of domain experts selected from different institutes for inspiring ideas, have been proved helpful for decision-making in various communities (Ruser, 2018), including healthcare (Goodolf & Godfrey, 2020), academy (Planells-Artigot, Ortigosa-Blanch, & Martí-Sánchez, 2021), enterprise management (Arshed, 2017), etc. In order to form a think tank, it is necessary to retrieve experts' and their institutes' information from official websites. However, in the era of Web 2.0, the number of websites and webpages has been increasing rapidly, and there are no uniform rules to obey in different websites. These facts impose great difficulties on think tank construction. On the one hand, browsing thousands of webpages in different websites to build a single think tank is time-consuming and labor-intensive. On the other hand, manual selection can be subjective, such that the independence of a think tank will be potentially undermined, which is crucial to the function and quality (Hernando, Pautz, & Stone, 2018; McGann, 2020). Under this circumstance, an automatic webpage retrieval method is needed to obtain experts' and their institutes' information efficiently to form a think tank.

* Corresponding author.
*E-mail address:* jinjian.jay@bnu.edu.cn (J. Jin).

https://doi.org/10.1016/j.ipm.2021.102767
Received 8 April 2021; Received in revised form 15 September 2021; Accepted 15 September 2021
Available online 20 October 2021
0306-4573/© 2021 Elsevier Ltd. All rights reserved.

In order to address the problem, the idea of query by example (QBE) has been introduced to webpage retrieval community. Different from traditional retrieval methods called query by keyword (QBK), QBE takes a given example as a query and returns similar candidates from a target collection. Since manual filtering and high-quality queries are no longer needed, QBE can largely lower the demands on users' information literacy and has become an ideal substitute of QBK in various real-world applications, including image retrieval (Amato, Carrara, Falchi, Gennaro, & Vadicamo, 2020; Li, Yang, & Ma, 2021), talent searching (Ha-Thuc et al., 2017), cross-platform product matching (Li, Dou, Zhu, Zuo, & Wen, 2019; Li, Xu, Luo, & Lin, 2014), document retrieval (Lopez-Otero, Parapar, & Barreiro, 2019; Weng et al., 2011) as well as cross-lingual tasks (Roostaee, Sadreddini, & Fakhrahmad, 2020; Sarwar & Allan, 2020), etc. A series of studies focused on webpage similarity were conducted, trying to apply QBE in webpage retrieval, denoted as query by webpage (QBW), among which hyperlinks between webpages (Jeh & Widom, 2002; Lin, Lyu, & King, 2006, 2009) and HTML documents (Bohunsky & Gatterbauer, 2010; Bozkır & Sezer, 2014; Bozkir & Sezer, 2018; Gowda & Mattmann, 2016; Li, Yang, Chen, Yuan, & Liu, 2019; Takama & Mitsuhashi, 2005) were considered. Nevertheless, methods based on hyperlinks improperly estimate the similarity between webpages in different websites, due to the scarcity of cross-website links. Also, HTML is an evolving standard for WWW, and the version and coding rules in different websites are not unified, so the performance of HTML-based methods is not satisfactory either.

Intuitively, webpages are different from linked texts, since their structured appearance usually organizes a list of elements in various styles. Such appearance provides instructive information for users to decide whether there is a need to examine the webpage or divert to another one, which is helpful for many tasks involved with webpage retrieval. As for think tank construction, even though target webpages probably do not look alike in details, they usually follow common visual patterns, which makes them more distinctive from irrelevant ones. For instance, the webpage of expert list usually contains a few lines of brief introduction for each expert, including his/her name, expertise, publications, sometimes photos, etc. As for the webpage of institutes' introduction, it usually contains large sections of text and a few images. Considering the limitation of HTML-based methods, the pattern can only be extracted from webpage's snapshot as visual features. Both color descriptors (Law, Thome, Gançarski, & Cord, 2012; Lowe, 2004) and neural networks (Fan, Guo, Lan, Xu, Pang, & Cheng, 2017; van den Akker, Markov, & de Rijke, 2019; Zhao et al., 2019) have been utilized to extract webpages' visual features for retrieval. However, color descriptors mainly extract trivial textures from the snapshot and ignore the high-level semantics, while neural networks based approaches are sensitive to the detail and position of images in a webpage.

Besides, it is necessary to consider webpages' textual content at the meantime for filtering, since irrelevant webpages may have similar visual appearance. QBW based on textual features, like document clustering and classification, etc., is a downstream task of textual content representation, which means that the representation of webpage content is crucial to the performance of QBW (Wu, Kanoulas, & de Rijke, 2020). Traditional representation methods are usually based on bag of words (BOW) model, due to the simplicity and robustness (Dourado, Galante, Gonçalves, & da Silva Torres, 2019; Jun, Park, & Jang, 2014; Kim, 2018; Lakshmi & Baskar, 2019; Pinheiro, Cavalcanti, & Tsang, 2017; Rinaldi, Russo, & Tommasino, 2020; Song, Liang, & Park, 2014; Weng et al., 2011; Wu et al., 2017; Yan, Li, Gu, & Yang, 2020). Compared to BOW-based methods, RNN structure and attention mechanism consider dependencies between terms as well as their relative importance (Chen et al., 2020; Jaeyoung, Janghyeok, Eunjeong, & Sungchul, 2020; Xu, Lin, Wu, & Wang, 2019). However, RNN-based methods generate feature vectors from the term grain and fail to capture the topical information, which is too precise for QBW (Lopez-Otero et al., 2019; Zhang, Li, & Wang, 2019). Moreover, proper nouns in a webpage can lower the estimated similarity score between relevant webpages from different domains, including persons' name, titles, technical terms, etc., which is ignored by existing extractors.

The aim of this study is to propose an automatic model to retrieve webpages similar to a given example from a target website, which is designed to obtain renowned experts and their institutes introduction efficiently for think tank construction. Inspired by users' searching intuitions, webpages' visual and textual features are considered in QBW. Particularly, the research objectives are as follows:

RO 1: To extract visual features which are high-level semantics less affected by the noise effect of images' details and positions in webpages;

RO 2: To extract textual features in an appropriately coarser grain;

RO 3: To estimate webpage similarity, according to which candidate webpages are ranked.

For RO 1, a visual feature extraction module (VEM) for webpage snapshot is proposed. In this module, rather than trivial details, high-level features are extracted by a pre-trained spatial pyramid pooling network (SPP). The noise effect brought by images' details and positions is mitigated via image unifying and a novel pooling scheme based on color diversity. Note that, the word 'unify' in VEM means to replace images with pre-defined ones according to their category labels which are provided by a pre-trained fully convolutional network (FCN). As for RO 2, a textual feature extraction module (TEM) is proposed which focuses on features from both term and topic grains. Proper nouns are all categorized and replaced according to a series of pre-defined rules, to reduce their negative influence on cross-domain retrieval tasks. As for RO 3, a textual similarity measure based on feature bootstrapping (FBS) is proposed to improve the robustness of textual similarity estimation, because a certain number of irrelevant webpages share a small proportion of extremely similar features with the given example, which makes it hard to filter them out practically. Besides, an adaptive weighting scheme is proposed to combine visual and textual similarity as the overall similarity between a given pair of webpages.

The main contributions are listed as follows:

1. A webpage retrieval model is proposed for think tank construction, leveraging webpages' visual and textual features;
2. Novel feature extraction modules are proposed for webpages' snapshots and textual content;
3. A series of metrics are proposed to measure webpage similarity for QBW.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. The proposed QBW model is elaborated in Section 3. In Section 4, extensive experiments are conducted to evaluate the proposed model, comparing to a series of baselines. Section 5 illustrates both the theoretical and practical implications of the model. Section 6 concludes this study.

## 2. Related works

The probable solution for automatic think tank construction leveraging expert finding model is firstly reviewed in this section. Then, the problem in existing QBW models that mostly based on hyperlinks between webpages and HTML documents is discussed. Accordingly, visual and textual feature extractors are introduced, which is hoped to be helpful for QBW.

### *2.1. Think tank construction*

Since building a think tank manually is time-consuming and subjective, expert finding models have been applied to collect experts' information automatically, most of which can be categorized into two types, i.e., finding experts from academic social networks and from the Web (Rostami & Neshati, 2019).

In fact, both types of models face problems in think tank construction practically. Specifically, models based on the Web leveraged candidate experts' local information for recommendation, and other documents in the dataset were utilized to describe associated candidates (Liang, 2019; Liang & de Rijke, 2016). However, in practice, there are a large number of irrelevant webpages mixed up with candidate ones in an institute's website, and few of them are associated with candidate experts or can be used to describe them. As for models based on academic social networks, candidates were crawled from academic databases (Chen, Treeratpituk, Mitra, & Giles, 2013; Wu, Fan, & Yuan, 2021; Yan, Liu, Shi, Wang, & Guo, 2017) or websites about community question answering (CQA) (Dargahi Nobari, Neshati, & Sotudeh Gharebagh, 2020; Dehghan, Abin, & Neshati, 2020; Dehghan, Biabani, & Abin, 2019). Apart from personal information, relations between candidates, like citation, question answering, etc., were considered as well. Nonetheless, these methods fail to find experts with few publication, like skilled programmers, experienced clinical doctors, etc., or those who are specialized in disciplines without a CQA, like literature, journalism, psychology, etc., yet they are all potentially helpful for decision-making.

Therefore, a webpage retrieval model is needed for think tank construction, which is hoped to accommodate the environment of real-world websites and obtain experts' and institutes' information automatically.

### *2.2. From QBE to QBW*

There have been a series of studies trying to bridge the gap between QBE and webpage retrieval, denoted as QBW, and existing methods are usually based on hyperlinks or HTML documents.

Hyperlink-based methods consider webpages as nodes and links as edges. It is believed that there should be more than one link between similar webpages (Jeh & Widom, 2002; Lin et al., 2006, 2009). However, due to the scarcity of cross-website links, webpages' similarity scores cannot be passed to different websites. Thus, the performance of hyperlink-based methods drops in a series of real-world applications, including expert retrieval, similar product retrieval, etc., because it is inevitable to obtain webpages similar to the given example from different websites.

QBW methods leveraging HTML documents are based on a backbone hypothesis that the similarity between HTML documents is equivalent to that between corresponding webpages. Both structural (Gowda & Mattmann, 2016; Nguyen, Le, & Vinh, 2014) and layout features (Bohunsky & Gatterbauer, 2010; Bozkır & Sezer, 2014; Bozkir & Sezer, 2018; Takama & Mitsuhashi, 2005) have been extracted from HTML documents for similarity estimation. However, versions of HTML are generally not unified yet, and different programmers have their own coding habits when maintaining websites. Therefore, the backbone hypothesis of HTML-based QBW methods is not always true, which results in poor performance, especially for cross-website retrieval. As argued by Bozkir et al. the dependency on HTML documents should be kept as low as possible for QBW due to the diversity of HTML versions (Bozkır & Sezer, 2014).

To summarize, existing methods ignore the particularity of different websites, which degrades their performance. Therefore, general features that are valid across different websites should be considered. For instance, users seldom focus on HTML documents or links between webpages when they are searching. On the contrary, it is easy to filter out a large number of irrelevant webpages merely based on the visual appearance of webpages, especially when searching for webpages that have distinctive visual characteristics like experts' webpages; otherwise, the textual content can be considered at the meantime for selection.

### *2.3. Visual and textual features for QBW*

Considering that webpages' visual and textual features can help to improve the performance of QBW models, existing visual and textual feature extractors are reviewed in this subsection.

Note that, different from layout features defined from textual content or HTML tags heuristically, visual features are directly extracted from webpages' snapshots (Fan et al., 2017). For instance, color descriptors like SIFT (Lowe, 2004) were firstly applied to extract feature vectors from snapshots of webpages for webpage archiving (Law et al., 2012). In order to extract high-level visual features, CNN-based extractors have been applied in various applications like traditional QBK as well as sentiment analysis (Zhao et al., 2019). For example, Fan et al. combined visual features extracted by CNN with heuristic features to train an end-to-end QBK

model (Fan et al., 2017), and an LSTM structure was applied to simulate users' F-biased browsing pattern (Faraday, 2000). In order to reduce model's demand on training data, Akker et al. transferred pre-trained VGG-16 and ResNet-152 to extract visual features from snapshots (van den Akker et al., 2019).

However, QBW aims to search for webpages that contain similar content/objects with the given example, instead of those identical to the example, while visual feature extractors mentioned above focus too much on details, which can degrade QBW performance. On the one hand, existing visual feature extractors in QBW models fail to extract high-level features like main objects from the snapshot. Specifically, CNN-based visual feature extractor has not been widely used in QBW yet, and color descriptors usually ignore high-level features and focus too much on details like color distribution and textures which are meaningless for QBW. On the other hand, feature vectors generated by above extractors are sensitive to the noise effect of images in webpages. More attention should be paid to the main content of images, instead of their details or positions, to determine whether the webpage is relevant or not. For instance, when searching for experts' webpages to form a think tank, for each image in a webpage, users are more concerned with whether it is a photo of experts. On the contrary, details, like clothes, hairstyle, background color, etc., and relative positions of images should not be considered, because they can only lower the estimated similarity score between relevant webpages, which makes it harder to recall them.

Because irrelevant webpages may have similar visual appearance, it is necessary to browse and understand the textual content at the meantime for webpage retrieval. QBW based on textual features is a downstream task of text representation, which means the quality of representation is crucial to the retrieval performance (Wu et al., 2020). A series of representation methods have been conducted to refine BOW model. For instance, in order to alleviate high-dimensionality and sparseness problem of BOW, various dimension reduction methods were applied to transform the document-term matrix into low-dimensional vectors (Jun et al., 2014; Song et al., 2014; Weng et al., 2011). Additionally, more information was added into the representation, including term-wise similarity (Kim, 2018; Pinheiro et al., 2017; Zhao & Mao, 2018), term frequency (Lakshmi & Baskar, 2019) as well as external information from WordNet (Rinaldi et al., 2020) and Wikipedia (Wu et al., 2017). In order to consider terms' semantics as well as dependencies between them, neural networks have been applied as textual feature extractors. For instance, Jaeyoung et al. applied doc2vec model (Le & Mikolov, 2014) for patent document retrieval (Jaeyoung et al., 2020). In addition, attention mechanism was applied with bi-LSTM to model the relative importance of terms (Xu et al., 2019). As for similarity metrics, siamese neural networks were applied in the semantic text matching model proposed by Jiang et al. (2019). To reduce the demand on training data of neural networks, pre-trained models like BERT were transferred to represent textual data (Chen et al., 2020; Wang & Zhang, 2020).

However, even though a series of studies have been conducted to refine BOW, such representation methods are still limited by discrete terms and fail to capture textual features from different semantic levels. Webpages usually contain long-form textual content, and local features captured by BOW are inadequate to represent the complex semantics in them, which can seriously degrade the performance of downstream QBW models. As for methods based on neural networks, features are extracted from the grain of terms, and these methods fail to capture topic distribution of the text, which is too precise for QBW (Lopez-Otero et al., 2019; Zhang et al., 2019). For instance, due to the ubiquity of synonym and polysemy, it is possible that a pair of webpages with different sets of exact terms share common topics, especially when they are in different websites or domains. They should be relevant to each other in retrieval tasks, but their textual similarity is lower estimated unexpectedly if features are extracted from the term grain only.

## 3. Proposed method

The proposed QBW model is illustrated in this section. The framework of the model is introduced in the first subsection. Also, the problem is defined, and notations used in this study are clarified. Then, components of the framework are elaborated, including visual and textual feature extraction modules and a series of similarity measures.

### 3.1. Framework

To form a think tank efficiently, a QBW model is proposed to retrieve experts' and their institutes' information from official websites, and the framework is presented in Fig. 1. Specifically, the model takes an exemplary webpage $p_0$ and a website $w = \{p_1, p_2, \dots, p_n\}$ as the input, and webpages similar to the example are returned as the output. The proposed model consists of visual and textual feature extraction modules (VEM and TEM) as well as corresponding similarity metrics which are introduced as follows.

Given a webpage $p_i$, visual and textual features are extracted via VEM and TEM separately. Particularly, VEM takes the URL of $p_i$ as the input and extracts a visual feature vector $\mathbf{v}_i = (\mathbf{v}_i^I, \mathbf{v}_i^T)$ from the snapshot $img_i$ of $p_i$, in which $\mathbf{v}_i^I$ and $\mathbf{v}_i^T$ represent visual feature vector of image-dominated and text-dominated regions in $p_i$ respectively. Representative webpage snapshots are presented in Fig. 2, including three expert lists and institute introduction pages. Note that, for privacy concerns, experts' photos and given names have been masked in Figs. 2 and 3. As for textual features, the textual content $txt_i = (txt_i^S, txt_i^N)$ of $p_i$ is taken as the input of TEM, and a textual feature vector $\mathbf{t}_i = (\mathbf{t}_i^S, \mathbf{t}_i^N)$ is returned, where $txt_i^S$, $txt_i^N$ denote semi-structured and non-structured texts in $p_i$, and $\mathbf{t}_i^S, \mathbf{t}_i^N$ represent corresponding feature vectors.

Afterwards, for a given pair of webpages, i.e., the example webpage $p_0$ provided by users and a candidate webpage $p_i$ in the website $w$, their similarity score is calculated via respective metrics. To be specific, paired feature vectors, $(\mathbf{v}_0, \mathbf{v}_i)$ and $(\mathbf{t}_0, \mathbf{t}_i)$, are taken as the input, and corresponding similarity scores, $sim_V(p_0, p_i)$ and $sim_T(p_0, p_i)$, are returned. Based on the adaptive weighting scheme, the overall similarity score $sim(p_0, p_i)$ is estimated as the weighted sum of $sim_V(p_0, p_i)$ and $sim_T(p_0, p_i)$. Note that, if both visual and textual features of webpages are available, the proposed VEM and TEM can be combined via the adaptive weighting scheme; otherwise, both modules can be applied alone.

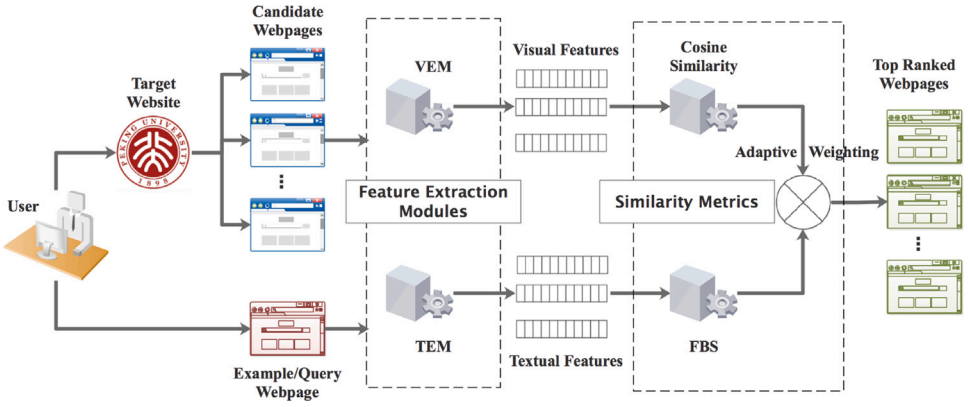



**Fig. 1.** Query by webpage.


**Fig. 2.** Examples of webpage snapshots.

### 3.2. Visual feature extraction module

Human beings can filter out a large number of irrelevant webpages merely based on their visual appearance, which implies that visual features can be helpful for webpage retrieval. Accordingly, a three-step visual feature extraction module is proposed.

#### 3.2.1. Image categorization and replacement

Images in a webpage provide users with rich information that are helpful for their comprehension about the webpage, and users will subconsciously filter out images' details and focus on the main content. For instance, given $p_0$ in Fig. 3 as the exemplary webpage for expert retrieval, users can easily tell $p_+$ is similar to $p_0$ while $p_-$ is not. Because they can recognize that the images in $p_+$ are experts' photos and ignore details like clothes, hair styles and background colors, so as those in $p_-$. In other words, users browse $p_0$, $p_+$ and $p_-$, while $p_0\prime$, $p\prime+$ and $p\prime-$ are actually in their minds. Therefore, to imitate such consideration, the proposed VEM firstly unifies webpages' images according to their category labels.

Specifically, a pre-trained FCN model is applied to provide category labels for image unifying. Note that, FCN is an open architecture comprising replaceable convolutional networks, and, with pre-trained on VOC dataset, an FCN-ResNet-101 model is applied in this study, which is constructed by an FCN model with a ResNet-101 backbone. 21 category labels are considered in the

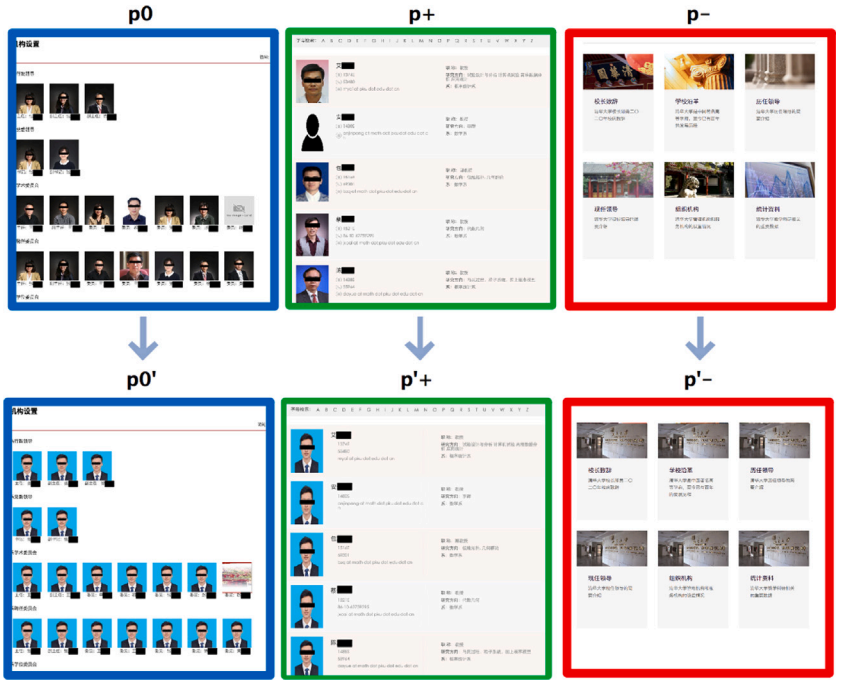



**Fig. 3.** A toy example of image categorization and replacement.

VOC dataset, including *background*, *aeroplane*, *bicycle*, *bird*, *boat*, *bottle*, *bus*, *car*, *cat*, *chair*, *cow*, *dining table*, *dog*, *horse*, *motorbike*, *person*, *potted plant*, *sheep*, *sofa*, *train*, *television* (Shelhamer, Long, & Darrell, 2017). In this study, they are merged into 3 categories, namely *person*, *background*, *other objects*, such that objects relevant to think tank construction can be identified correctly, like person, building, etc. It is worth noting that the recognition result of FCN can affect the following feature extraction, because FCN determines how the images are replaced. Fortunately, the performance of FCN is promising ($accuracy = 90.3\%$ on VOC dataset) (Shelhamer et al., 2017), so its influence is believed to be limited.

In order to show the impact of image unifying, a pre-trained SPP model, which is used to extract visual features, and the cosine function are applied to roughly estimate the similarity between $p_0$ and $p+$, $p-$ in Fig. 3. Specifically, $cos\left(p_0, p+\right) = 0.62$, $cos\left(p_0, p-\right) = 0.58$, while $cos\left(p_0\prime, p\prime+\right) = 0.72$, $cos\left(p_0\prime, p\prime-\right) = 0.55$. It implies that the positive and negative samples become more distinctive after images are unified, which can potentially improve the performance of QBW.

### 3.2.2. Feature vector extraction

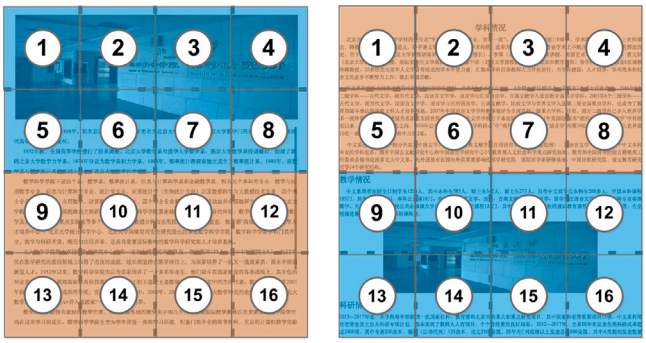
Once images are unified, a snapshot is generated for each webpage for visual feature extraction. It is worth noting that snapshots are converted into gray scale to mitigate the noise effect of color. In practice, generated snapshots are large in scale and random in aspect ratio, while most extractors can only process fixed-sized images, due to the structure of the fully connected layers within them, e.g., VGG-16 can only process images sized $224 \times 224$. Thus, snapshots need to be cropped and warped into required sizes. However, over-resizing will blur the snapshot and lose informative features, which badly affects the quality of the extracted vectors.

To address this problem, a pre-trained SPP model is applied to extract visual feature vector $\mathbf{v}_i$ from snapshot $img_i$ of $p_i$. The SPP model comprises a replaceable visual feature extractor and a spatial pyramid pooling layer, and it can process images of arbitrary scale and aspect ratio and generate feature vectors of a fixed dimension with no need of resizing the input image (He, Zhang, Ren, & Sun, 2015).

It is believed that the semantic information is contained in high-layers of CNNs, while details like textures are in low-layers which can bring noise effect to the similarity estimation (Yu, Yang, Yao, Sun, & Xu, 2017). Therefore, pre-trained on ImageNet dataset, 13 convolutional layers that share the same structure with those in VGG-16 model are applied as the feature extractor in SPP, different from the original architecture where only 5 layers are used (He et al., 2015). Then, the spatial pyramid pooling layer in SPP divides the generated feature maps into a fixed number of regions, each of which matches the corresponding region in the original snapshot, and max pooling is conducted on each region of the feature map. Finally, a feature vector is obtained with the size of $d \times a \times a$. $d$ denotes the number of feature maps determined by the number of kernels in the feature extractor, which is 512 due to the structure of the 13 convolutional layers, and $a \times a$ means the number of regions divided by SPP which is set by users.

Mathematically, regional visual feature vectors for webpage $p_i$ can be denoted as

$$\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, \ldots, \mathbf{v}_i^{(a\times a)} = SPP\left(img_i\right) \tag{1}$$




(a) Image at the top (b) Image in the text

**Fig. 4.** A toy example of pooling scheme based on color diversity.

### 3.2.3. Feature vector pooling

In practice, images and texts are usually placed in different positions of a webpage. As shown in Fig. 4, both webpages are college introduction page with image unified but placed in different positions. If the regional feature vectors are concatenated directly for further comparison, the visual similarity will be lower estimated.

Accordingly, a pooling scheme based on color diversity is proposed, which helps to recognize whether a region is dominated by images or texts, and feature vectors can be pooled separately. In this way, for a given pair of webpages, regions dominated by images in each webpage can be matched and compared, so as the text-dominated ones. For instance, in Fig. 4, region 1 to 8 in Fig. 4(a) should be considered as image-dominated and matched with region 9 to 16 in Fig. 4(b), so as the rest ones.

Specifically, for a webpage, its original snapshot with three channels is firstly divided into $a \times a$ regions, consistent with the output of SPP model. Then, the color diversity for $j$th region in the original snapshot is calculated as $div^{(j)} = \|\mathbf{R}^{(j)} - \mathbf{G}^{(j)}\|_2^2 + \|\mathbf{B}^{(j)} - \mathbf{G}^{(j)}\|_2^2 + \|\mathbf{R}^{(j)} - \mathbf{B}^{(j)}\|_2^2$, in which $\mathbf{R}^{(j)}, \mathbf{G}^{(j)}, \mathbf{B}^{(j)}$ denote the pixel matrix of the three channels in $j$th region respectively. Afterwards, regions are ranked according to their $div$ values, and the top ranked half regions are considered as image-dominated, while the rest ones are text-dominated. Note that, the proposed color diversity function $div$ does not aim to calculate the richness of color for snapshots, but it is an effective way to distinguish images and texts. Finally, average pooling is conducted on feature vectors of image-dominated and text-dominated regions, and $\mathbf{v}_i^I$, $\mathbf{v}_i^T$ are generated separately. The global visual vector $\mathbf{v}_i$ is the concatenation of $\mathbf{v}_i^I$ and $\mathbf{v}_i^T$, which helps to preserve rich information in both images and texts. In this way, images and texts in different webpages can be matched for comparison wherever they are placed in the webpage. Given the snapshot $img_i$ of a webpage $p_i$, the proposed pooling scheme can be denoted as

$$\mathbf{v}_i^{(r_1)}, \mathbf{v}_i^{(r_2)}, \ldots, \mathbf{v}_i^{(r_{a\times a})} = rank\left([\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, \ldots, \mathbf{v}_i^{(a\times a)}], [div_i^{(1)}, div_i^{(2)}, \ldots, div_i^{(a\times a)}]\right)$$

$$\mathbf{v}_i^I = \frac{1}{a\times a/2}\sum_{r_j=1}^{a\times a/2}\mathbf{v}_i^{(r_j)} \quad \mathbf{v}_i^T = \frac{1}{a\times a/2}\sum_{r_j=a\times a/2+1}^{a\times a}\mathbf{v}_i^{(r_j)} \tag{2}$$

$$\mathbf{v}_i = \left(\mathbf{v}_i^I, \mathbf{v}_i^T\right)$$

in which $div_i^{(j)}$ represents the color diversity of $j^{th}$ region in $img_i$, $rank\left(\xi_1, \xi_2\right)$ means ranking elements in $\xi_1$ according to the corresponding values in $\xi_2$, and $\mathbf{v}_i^{(r_j)}$ denotes the $r_j^{th}$ ranked regional feature vector.

Note that, $a\times a/2$ is set as the margin index between regions dominated by images or texts, which helps to leverage the number of regions occupied by images and texts for similarity estimation. Particularly, if images take up far more than $a\times a/2$ regions in $p_i$, there will be feature vectors of images mixed up into $\mathbf{v}_i^T$, and vice versa. Thus, the estimated visual similarity between webpages that contain different number of images will decrease, which makes them more distinctive. For instance, $p\prime-$ in Fig. 3 and the webpage in Fig. 4(b) can be easily distinguished, even though their images have been unified, because $p\prime-$ contains much more images and less texts compared to the webpage in Fig. 4(b).

## 3.3. Textual feature extraction module

Even though webpages' visual appearance helps users filter out obviously irrelevant webpages, the visual similarity is not completely equivalent to the webpage similarity. Hence, it is still necessary to read and understand the textual content at the meantime. Accordingly, a textual feature extraction module is proposed.

**Table 1**
Examples of term replacement rules.

| Original term | POS label | Description | Replaced term |
|---|---|---|---|
| J.Z. Zhang | *nr* | Name of a person | Person |
| Associate professor | *nnt* | Title | Title |
| Calculus | *gm* | Mathematical term | Academic term |
| Quantum mechanics | *gp* | Physical term | Academic term |

#### 3.3.1. Term recognition and replacement

There are a large number of proper nouns in institutes' websites, which makes the extracted features overly precise for QBW and lowers the estimated similarity between relevant webpages with different exact terms, especially in general tasks like cross-discipline and cross-website retrieval.

In order to alleviate this problem, proper nouns in a webpage are recognized and replaced via POS tagging. Specifically, a series of replacing rules are defined according to their POS labels, a few of which are listed in Table 1. In fact, term recognition and replacement helps TEM extract coarse-grained features from the textual content, similar to image categorization and replacement in VEM. In this way, users will no longer be required to provide exemplary webpages that are from the same domain with the target website.

#### 3.3.2. Feature vector extraction

After proper nouns are recognized and replaced, textual features are generated from the textual content $txt_i$ of each webpage $p_i$. In practice, there are always semi-structured texts $txt_i^S$ like tables and non-structured texts $txt_i^N$ like paragraphs in one webpage at the meantime. Considering that both semi-structured and non-structured texts have their unique characteristics, two types of texts are distinguished, and different approaches are applied for feature extraction subsequently. Intuitively, sentences in non-structured texts usually have a complete syntactic structure, while those in semi-structured texts do not, so syntactic parsing is conducted for sentences in a specific webpage to distinguish them. Particularly, if there are any *subject-verb* relation in a sentence, it is categorized as non-structured text, otherwise it is semi-structured.

Semi-structured texts usually comprise discrete terms with little dependency or connection between them, so a representation method ignoring relations between terms should be applied for semi-structured texts. Moreover, terms in semi-structured texts like tables usually carry critical semantics, which makes each webpage more distinctive. Accordingly, average word vector generated by GloVe model is utilized, because word vectors can preserve rich semantics of terms compared to models based on BOW, which is helpful for the following similarity estimation.

As for non-structured texts, it is worth noting that QBW aims to retrieve webpages similar or relevant to the given example in the topic level, instead of identical ones in the term level. However, similar topics can be expressed with different exact terms due to the ubiquity of synonym and polysemy. Thus, their similarity score can be lower estimated unexpectedly if textual features are extracted from the term grain only, which makes it harder to retrieve topic relevant pages. Accordingly, the topic distribution is extracted from non-structured texts via an LDA model, which is hoped to be an appropriately coarse grain to extract features for QBW. Finally, feature vectors of semi-structured and non-structured texts are concatenated as the global textual feature vector $\mathbf{t}_i$ for a particular webpage $p_i$. It can be denoted as

$$\begin{aligned} \mathbf{t}_i^S &= \frac{1}{n_i^S} \sum_{r^{(j)} \in txt_i^S} \mathbf{r}^{(j)} \\ \mathbf{t}_i^N &= LDA\left(txt_i^N\right) \\ \mathbf{t}_i &= \left(\mathbf{t}_i^N, \mathbf{t}_i^S\right) \end{aligned} \tag{3}$$

in which $n_i^S$ denotes the number of terms within $txt_i^S$, and $\mathbf{r}^{(j)}$ represents the GloVe vector of term $r^{(j)}$.

For the generated visual and textual feature vector of a given pair of webpages, i.e., an example webpage $p_0$ and a candidate webpage $p_i$, a series of metrics are proposed to measure the similarity between them which are elaborated in the following subsections.

### 3.4. Visual similarity

Considering the high-dimensionality of VEM's output vector, cosine function is applied as the similarity metric for the visual feature vector $\mathbf{v}_0$ and $\mathbf{v}_i$ of $p_0$ and $p_i$. Cosine function regards the similarity between a pair of vectors as the angle cosine between them, which mitigates the bad influence due to the curse of dimensionality. Practically, cosine function has been applied for image matching in existing works and achieves satisfying performance compared with other metrics (Snell, Swersky, & Zemel, 2017; Vinyals, Blundell, Lillicrap, Kavukcuoglu, & Wierstra, 2016).

$$sim_V\left(p_0, p_i\right) = cos\left(\mathbf{v}_0, \mathbf{v}_i\right) = \frac{\mathbf{v}_0 \cdot \mathbf{v}_i}{\|\mathbf{v}_0\| \cdot \|\mathbf{v}_i\|} \tag{4}$$

$sim_V\left(p_0, p_i\right)$ denotes the visual similarity between webpages $p_0$ and $p_i$

### 3.5. Textual similarity

In practice, there are a certain number of irrelevant webpages whose textual feature vectors share a small proportion of features that are extremely similar. For instance, webpages about notifications of academic reports may share some topics with that about the introduction of an institute. Thus, to alleviate such problem, a novel similarity metric is needed which helps elude extremely similar features that possibly exist. Otherwise, the similarity score can be higher estimated.

Accordingly, for the textual feature vectors $\mathbf{t}_0$ and $\mathbf{t}_i$ of $p_0$ and $p_i$, the proposed FBS samples a certain proportion of features from each vector and calculates the cosine value based on the sampled features at a time, which is repeated for $\eta$ rounds. Finally, the average value of cosine similarity in each round is regarded as the textual similarity between $p_0$ and $p_i$. It can be denoted as

$$\mathbf{t}_0^{(j)} = FBS\left(\mathbf{t}_0, \delta, \tau^{(j)}\right) \quad \mathbf{t}_i^{(j)} = FBS\left(\mathbf{t}_i, \delta, \tau^{(j)}\right)$$

$$sim_T\left(p_0, p_i\right) = \frac{1}{\eta}\sum_{j=1}^{\eta} cos\left(\mathbf{t}_0^{(j)}, \mathbf{t}_i^{(j)}\right) = \frac{1}{\eta}\sum_{j=1}^{\eta} \frac{\mathbf{t}_0^{(j)} \cdot \mathbf{t}_i^{(j)}}{\|\mathbf{t}_0^{(j)}\| \cdot \|\mathbf{t}_i^{(j)}\|} \tag{5}$$

in which $\mathbf{t}_0^{(j)}$ and $\mathbf{t}_i^{(j)}$ denote the sampled features from $\mathbf{t}_0$ and $\mathbf{t}_i$ in $j$th round, $\tau^{(j)}$ represents the random seed utilized in $j$th round sampling, and $\delta$ denotes the sampling proportion which is set by users.

### 3.6. Adaptive weighting scheme

Intuitively, the discrimination of webpages' visual features differs in each website. For instance, if most webpages in one website contain a fixed number of images, and the noise effect of images has been alleviated by VEM, their visual appearance can be considered as similar to each other via feature vectors. Under such circumstance, visual features have little help for distinguishing webpages in the website, which means a small weight should be given to the visual similarity for better discrimination. Hence, it is necessary to combine webpages' visual and textual similarity via an adaptive weighting scheme, and the distinction of the number of webpages' images in a website may mirror visual features' ability to distinguish webpages to some extent.

Accordingly, for a target website $w$, the range of the number of images in $w$, denoted as $range(w)$, is leveraged to calculate the relative weight of visual and textual similarity for QBW, which is defined as

$$\alpha(w) = \max\left(0, 1 - \frac{1}{\ln(range(w))}\right) \tag{6}$$

Considering that the range of the number of images is a discrete variable, a logarithmic function is applied to map it into a consistent space. Moreover, due to the characteristic of logarithmic function, the growth rate of $\alpha(w)$ slows down as $range(w)$ increases. In this way, $\alpha(w)$ is limited into a reasonable interval, in case that $range(w)$ gets extremely big.

Apart from the proposed adaptive weighting scheme, a fixed pair of weights can be determined via sensitivity analysis which is evaluated in Section 4.5.1.

For a given exemplary webpage $p_0$ and $\forall p_i \in w$, the overall similarity between $p_0$ and $p_i$ is defined as

$$sim\left(p_0, p_i\right) = \alpha(w) \times sim_V\left(p_0, p_i\right) + (1 - \alpha(w)) \times sim_T\left(p_0, p_i\right) \tag{7}$$

Finally, candidate webpages $p_i$ in the target website $w$ are ranked according to their overall similarity score $sim\left(p_0, p_i\right)$ with the given example $p_0$, and the top ranked ones are returned to users.

## 4. Experiments

In this section, the proposed QBW model is evaluated on two datasets, compared with a series of baselines. Particularly, four webpage retrieval tasks related to think tank construction are focused, including experts' and institutes' information retrieval in academic and medical scenarios.

### 4.1. Datasets

Considering that QBW is still an emerging research question, and there is no benchmark datasets with golden-label, two datasets for webpage retrieval were collected and labeled manually for evaluation. Descriptions of datasets are shown in Table 2.

Specifically, to form a think tank, websites of academic institutes are usually the first choice, because experts' information is sufficient and regularly maintained. Thus, it is necessary to evaluate models' performance in academic scenario, so 12 top ranked colleges' websites in China were crawled, 10241 webpages in total, denoted as Dataset 1. Besides, think tanks have already been applied in medical community, so model's performance still matters in medical scenario. What is more, webpages in hospitals' websites are different from those in colleges' websites in various aspects, e.g., text expressions, page layouts, resources, etc., so the performance in medical scenario can prove the versatility of the proposed model. Then, 7 websites of famous hospitals in China were also crawled, 12340 webpages in total, denoted as Dataset 2. For each webpage, textual content and resources like HTML document and images were crawled, and a snapshot was generated after images were unified. In order to evaluate models' cross-domain retrieval performance, webpages from various domains were collected, including mathematics, computer science,

**Table 2**
Descriptions of datasets.

| Dataset | Size | Unit | Scenario | Task |
|---|---|---|---|---|
| Dataset 1 | 10241 | webpage | academic | Task 1: scholar retrieval<br>Task 2: college introduction retrieval |
| Dataset 2 | 12340 | webpage | medical | Task 3: doctor retrieval<br>Task 4: medical section introduction retrieval |

**Table 3**
Exemplary webpages.

| Task | URL of Exemplary webpage |
|---|---|
| Task 1 | URL 1: http://math.bnu.edu.cn/jzg/szdw/index.htm<br>URL 2: http://psych.bnu.edu.cn/tabid/50/Default.aspx |
| Task 2 | URL 3: https://www.math.pku.edu.cn/xygk/yzzc/index.htm<br>URL 4: https://www.im.pku.edu.cn/zjxg/xggk/index.htm |
| Task 3 | URL 5: http://www.pkuph.cn/Article/Index/department/id/84/cid/84/type/2/<br>URL 6: http://anzhen.org/Html/Departments/Main/SearchIndex_224.html |
| Task 4 | URL 7: http://www.xhbnet.com/ksgc/index_16.aspx<br>URL 8: http://www.whuh.com/sections/detail/id/4/select/desc.html |

information management, literature in Dataset 1, and emology, endocrinology, neurology in Dataset 2. Additionally, exemplary webpages from different domains were used in each retrieval task.

Dataset 1 and 2 are of high quality in terms of size as well as heterogeneity. Because the number of webpages in each website is similar, between 1000 and 2000 to be specific, it is reasonable to assume that the retrieval task on each website is a duplicated work, i.e. to retrieve target webpages from a candidate set sized 1000 to 2000 webpages. It means that the task needs to be repeated for 12 times in academic scenario and 7 times in medical scenario, due to the current size of Dataset 1 and 2. Therefore, it can be assumed that the two datasets, with all webpages from 12 and 7 websites, are qualified for the evaluation of different algorithms. Apart from sufficient size, each dataset contains webpages with heterogeneous visual appearance as well as textual content. As shown in Fig. 2, some webpages contain a certain number of images, including photos of experts, the facade of buildings, etc., like expert list 1, 2 and institute introduction 2 in Fig. 2, while others do not, like expert list 3 and institute introduction 1, 3 in Fig. 2. As for textual content, webpages usually contain semi-structured and non-structured texts at the meantime, both of which have their own features, like the paragraph and table in institute introduction 3 in Fig. 2. The heterogeneity of Dataset 1 and 2 helps to evaluate models in terms of their robustness as well as generality.

Note that, the four retrieval tasks can be classified into two categories, i.e., expert and institute introduction retrieval. Expert retrieval is the prior concern to form a think tank, and more importantly, it can also be considered as a representative of tasks that search for webpages following specific patterns to some extent, like leaders of institutes, curriculums of colleges, schedules of doctors, etc. On the contrary, institute introduction retrieval represents tasks aimed for general webpages in a website, e.g., the introduction, history and news of an institute. Thus, it is expected that the two kinds of tasks can prove not only the proposed model' s potential for think tank construction, but its versatility on other general tasks as well.

According to the particular task, all webpages $p_i$ in Dataset 1 and 2 were labeled as $y_i \in \{0, 1, 2\}$ and double-checked by two postgraduates majored in information science, and disputed ones are further discussed to finalize their labels. Particularly, *label 2* means the webpage is completely relevant to the given example, and *label 1* means that only some parts of the webpage are relevant, while *label 0* is given to irrelevant webpages. For instance, when searching for scholars' webpages in a college's website, all scholars' webpages in the website were labeled as 2, and webpages that involved with scholars' information like curriculum and syllabus were labeled as 1, while completely irrelevant webpages like news were labeled as 0. It is worth noting that websites in Dataset 2 contain much more target webpages ($y_i > 0$) than those in Dataset 1. To be specific, there are an average of 8.8 target webpages per website in Dataset 1 for scholar retrieval, 7 for college introduction retrieval, while 23 in Dataset 2 for doctor retrieval, and 26.3 for medical section introduction retrieval. As for dataset division, 20% of webpages in Dataset 1 and 2 were randomly sampled as corresponding validation dataset for sensitivity analysis to determine the value of hyper-parameters.

A series of example webpages from different colleges/hospitals and domains/sections were selected for each task. A few instances are listed in Table 3.

### 4.2. Evaluation metrics

In this study, precision@k (k=1,5,10), recall@k (k=5,10), NDCG@k (k=5,10) and mean average precision are utilized to evaluate models' performance, denoted as P@k, R@k, N@k and MAP for short respectively.

Mathematically, P@k and R@k are defined as

$$P@k = \frac{TP@k}{k} \qquad R@k = \frac{TP@k}{TP + FN} \tag{8}$$

in which TP and FN mean the number of true positives and false negatives respectively, and TP@k denotes the number of true positives among the top-ranked k webpages.

N@k can be defined as

$$N@k = \frac{DCG@k}{IDCG@k}$$

$$DCG@k = \sum_{i}^{k} \frac{2^{r(i)} - 1}{\log_2 (i+1)} \tag{9}$$

in which $r(i)$ represents the relevance score of the $i$th ranked webpage, and IDCG@k means the DCG@k score of an ideally ranked list.

MAP is defined as

$$MAP = \frac{1}{|Q|} \sum_{p_0 \in Q} AP\left(p_0\right)$$

$$AP = \frac{\sum_k y_k \times P@k}{\sum_k y_k} \tag{10}$$

in which Q denotes the set of example webpages, and AP represents the average precision value calculated using $\forall p_0 \in Q$ as the example.

### 4.3. Baselines

**VEM+TEM+FBS+$\alpha^*$**: The proposed model. To be specific, the proposed VEM and TEM are used to extract visual and textual features of webpages. FBS is proposed as a novel textual similarity metric. Adaptive weighting scheme balances the effect of visual and textual features on webpage similarity, denoted as $\alpha^*$ for convenience.
**VEM+TEM+FBS+$\alpha^0$**: $\alpha^0$ means that weights for visual and textual features are fixed and determined via sensitivity analysis.
**VEM+TEM+$\alpha^*$**: Textual similarity is calculated via cosine function instead of FBS.
**VEM+LDA+$\alpha^*$**: Textual features of webpages are extracted by a single LDA model.
**VEM+LDA+$\alpha^0$**: Textual features of webpages are extracted by a single LDA model, and weights for visual and textual features are fixed and determined via sensitivity analysis.
**VEM+BERT+$\alpha^*$**: Textual features of webpages are extracted by a pre-trained BERT model.
**VEM+BERT+$\alpha^0$**: Textual features of webpages are extracted by a pre-trained BERT model, and weights for visual and textual features are fixed and determined via sensitivity analysis.
**TEM+FBS**: Only textual features extracted by TEM are used for QBW.
**VEM**: Only visual features extracted by VEM are used for QBW.
**VGG-16**: Only visual features extracted by VGG-16 are used for QBW.
**MobileNet**: Only visual features extracted by MobileNet are used for QBW. MobileNet is a CNN-based image classifier, and it can be transferred to extract visual features for general computer vision tasks. Due to depth-wise separable convolutions, MobileNet is much lighter than other CNN-based models in computer vision scenarios (Howard et al., 2017).
**$\epsilon$-ACSM**: $\epsilon$-ACSM is a patch-based image similarity measure. It calculates the similarity as the average area of matched patches between two original images. 2D Hamming distance is applied for approximate patch matching in a given neighborhood of the patch, to alleviate the problem of exact patch matching (Amelio, 2019).
**LDA**: Only textual features extracted by LDA are used for QBW.
**Q-BERT**: Only textual features extracted by BERT are used for QBW (Wang & Zhang, 2020).
**HTML Similarity**: Webpages' HTML documents are transformed to DOM trees, and edit distance is applied to measure the similarity between webpages (Gowda & Mattmann, 2016).
**URL2vec**: URL2vec is a URL representation learning model based on temporal convolution neural networks. Features are learnt by predicting tokens in the URL given their previous $k$ tokens (Liang et al., 2019).
**Kazemian's**: A series of lexical features of URLs are applied, including length of URLs, special symbols, n-gram, tf-idf, etc. K-means is utilized to distinguish target webpages from candidate ones (Kazemian & Ahmed, 2015).

### 4.4. Parameter settings

Specific parameter settings in the proposed model as well as baselines are presented in this subsection. Other parameters are set based on a series of sensitivity analyses which are shown in the following subsection.

Most parameters are associated with VEM, and their values are shown in Table 4.

Specifically, 13 convolutional layers are used in VEM to extract high-level features. Due to the structure of the convolutional layers, the dimension of regional vectors is $d = 512$, and that of concatenated global vectors is 1024. Although SPP model can process images of arbitrary scale and aspect ratio, the upper bound of generated snapshots' height is set as 1600 pixels, which is the height of two concatenated full-screen browser windows. Intuitively, useful information will not be placed at the bottom of a webpage,

**Table 4**
Parameter settings for VEM.

| Conv.layers | Dimension | Max height | Regions |
|---|---|---|---|
| 13 | 1024 | 1600 | 4 × 4 |

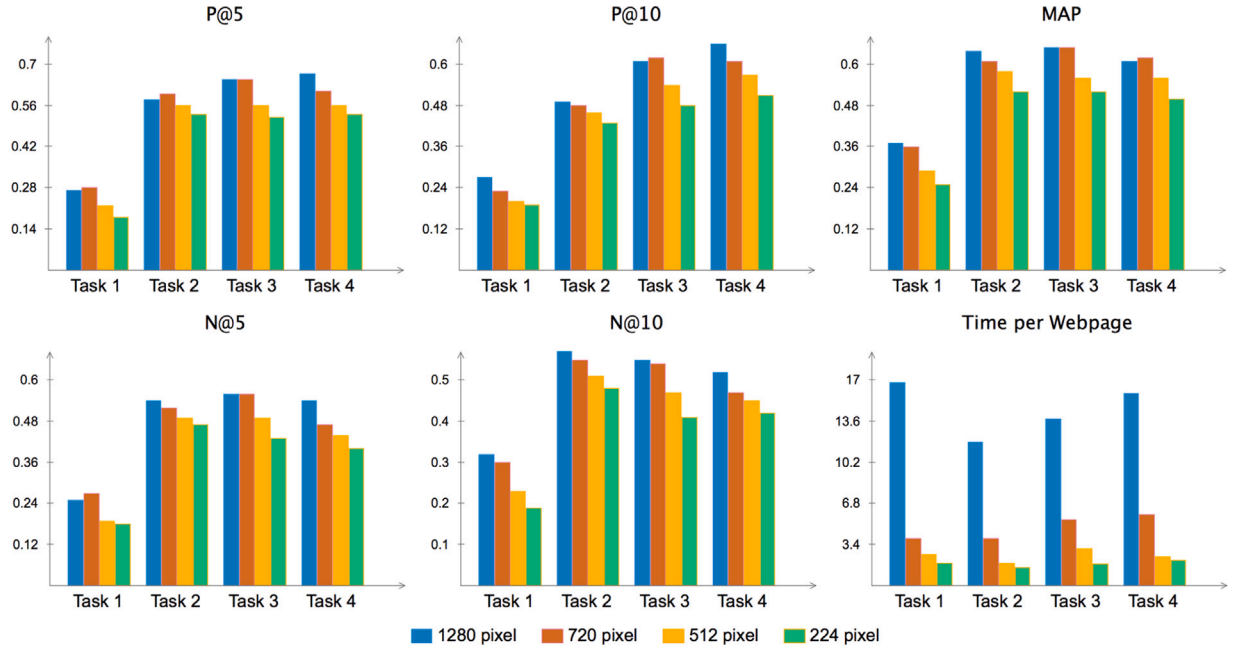



**Fig. 5.** Performance on different snapshot width.

and twice the screen height should be enough for users to scroll for browsing, considering users' searching habits. Moreover, based on parameter settings in the original architecture of SPP model (He et al., 2015), the number of regions divided in a snapshot is set as $4 \times 4$ ($a = 4$).

As for TEM, the feature vector for each webpage's semi-structured text is extracted via the GloVe model, and its dimension is set as 300.

### 4.5. Result analysis

#### 4.5.1. Sensitivity analysis

The influence of parameters in the proposed methods and baselines on the performance is analyzed in this subsection.

(1) The Scale of Snapshots

First of all, the scale of snapshots needs to be set, because it affects not only the retrieval performance, but also the time complexity of VEM. Considering that the input of SPP model can be of arbitrary aspect ratio, a series of sensitivity analyses on the width of snapshots were conducted, and the height can be calculated preserving the original aspect ratio. Specifically, the performance of VEM on 4 tasks based on different width was evaluated in terms of 5 metrics and the processing time per snapshot, as presented in Fig. 5. Task 1 to 4 in the horizontal axis denotes the four tasks listed in Table 2 respectively.

It can be seen from the histograms that switching the width of snapshots from 1280 to 720 pixels has little impact on the performance in each task, while it can largely reduce the time cost of VEM. Nonetheless, once the width of snapshots is compressed to 512 pixels or smaller, the average performance drops obviously, but the time cost is not reduced obviously. Thus, the width of webpages' snapshots is set as 720 pixels to make a trade-off between model's performance and efficiency, based on which the height of each snapshot can be calculated preserving the original aspect ratio.

(2) Abstract Images

Apart from concrete hyper-parameters, abstract images are sometimes used in institutes' websites, which may affect image unifying in VEM. Hence, the sensitivity of the overall performance of VEM to such influence is elaborately analyzed here. As shown in Fig. 6, abstract images can be roughly categorized into three types.

The first type of abstract image usually appears in navigation bars of websites as buttons with different functions, like that in Fig. 6(a). In practice, they are small in size and will not appear on a large scale in a webpage. Thus, their impact is believed to be

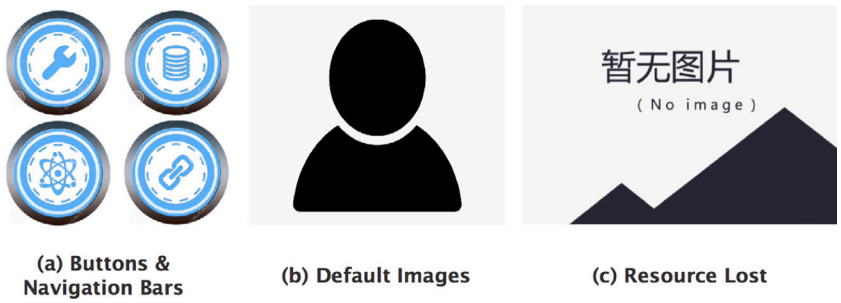



**Fig. 6.** Examples of three types of abstract images.

limited. Besides, default images are used in some websites for a consistent UI design, e.g., a default image for a person is presented in Fig. 6(b). As shown in $p_+$ and $p_+\prime$ in Fig. 3, such abstract images can be correctly recognized by the pre-trained FCN model and replaced with corresponding unified concrete images. Moreover, if some particular resource is lost, abstract images like that in Fig. 6(c) will be employed to inform users. It is demanding for FCN to label such images correctly, and they will much likely be replaced with a wrong image, as shown in $p_0$ and $p_0\prime$ in Fig. 3. However, the official website of an institute that is considered for think tank construction should be of high quality, so the loss of resources only occurs sporadically, not on a large scale. Hence, abstract images referring to resource loss will be in small proportion in a webpage, and most images will be concrete, which indicates that their negative effect is limited.

(3) The Number of Topics

Besides, a series of sensitivity analyses were conducted to determine the optimal number of topics in LDA, because it has an impact on model's overall performance, as presented in Fig. 7. It is shown that the number of topics which is too large or small can result in poor retrieval performance. Considering the generality of the model on different tasks, the number of topics in TEM as well as other baselines involved with LDA model is set as 300. For a webpage, the feature vector extracted from non-structured texts is concatenated with that from semi-structured texts which is generated via GloVe model. Thus, the dimension of the textual feature vector for each webpage is 600.

(4) The Number of Sampled Features

Also of concern is the selection of $\delta$ in FBS, which is the number of sampled features to measure textual similarity between webpages. Intuitively, too large the value of $\delta$ is hardly helpful for the robustness of textual similarity estimation, because most features will be sampled, while too small the value of $\delta$ can make the similarity estimation extremely unstable. Thus, to select a proper $\delta$, the retrieval performance based on different values of $\delta$ on each task was evaluated. As shown in Fig. 8, as $\delta$ increases from 100 to 300, the retrieval performance becomes better obviously, while the performance holds on expert retrieval tasks and even drops on institute introduction retrieval tasks as $\delta$ continues to increase from 300 to 600. Accordingly, the value of $\delta$ is set as 300.

As presented in Figs. 7 and 8, different number of topics and the value of $\delta$ have much more impact on the performance on institute introduction retrieval tasks, compared to that on expert retrieval tasks. It is probably resulted from the segmentation of semi-structured and non-structured texts. Particularly, there are richer non-structured texts in example webpages for institute introduction retrieval tasks, while the textual content of those for expert retrieval tasks is dominated by semi-structured texts, like lists of names, titles, expertise, etc. Thus, the performance on expert retrieval tasks is relatively stable to the changing number of topics and the value of $\delta$.

(5) The Weight of Visual and Textual Similarity

In addition, the performance of baselines that consider both visual and textual features was evaluated on each task, using different weights for each type of similarity. The weight of visual similarity is denoted as $\alpha$ for convenience, and that of textual similarity is $1-\alpha$. Specifically, the performance curves of VEM+TEM+FBS+$\alpha^0$, VEM+LDA+$\alpha^0$ and VEM+BERT+$\alpha^0$ on different tasks are presented in Fig. 9 to 11 respectively.

Under most circumstances, as $\alpha$ increases, the performance keeps improving until $\alpha$ reaches a specific value, then it begins to hold or even decline as $\alpha$ continues to increase. Thus, the specific value of $\alpha$ corresponding to the turning point in a curve should be selected. To make the performance of the three baselines more competitive, different values of $\alpha$ are selected for each of them. According to Figs. 9 and 11, $\alpha$ is set to be 0.6 in VEM+TEM+FBS+$\alpha^0$ and VEM+BERT+$\alpha^0$, while that in VEM+LDA+$\alpha^0$ is set to be 0.7 based on Fig. 10.

The trend of curves in Fig. 9 to 11 implies the positive effect of webpages' visual and textual features on webpage retrieval tasks for think tank construction, because too large or too small value of $\alpha$ will result in relatively poor performance. For instance, as the value of $\alpha$ decreases, which means the effect of visual features is ignored gradually, three models' performance on different tasks drops significantly. It is also consistent with the result analysis in Tables 5 to 8 in the following subsection.

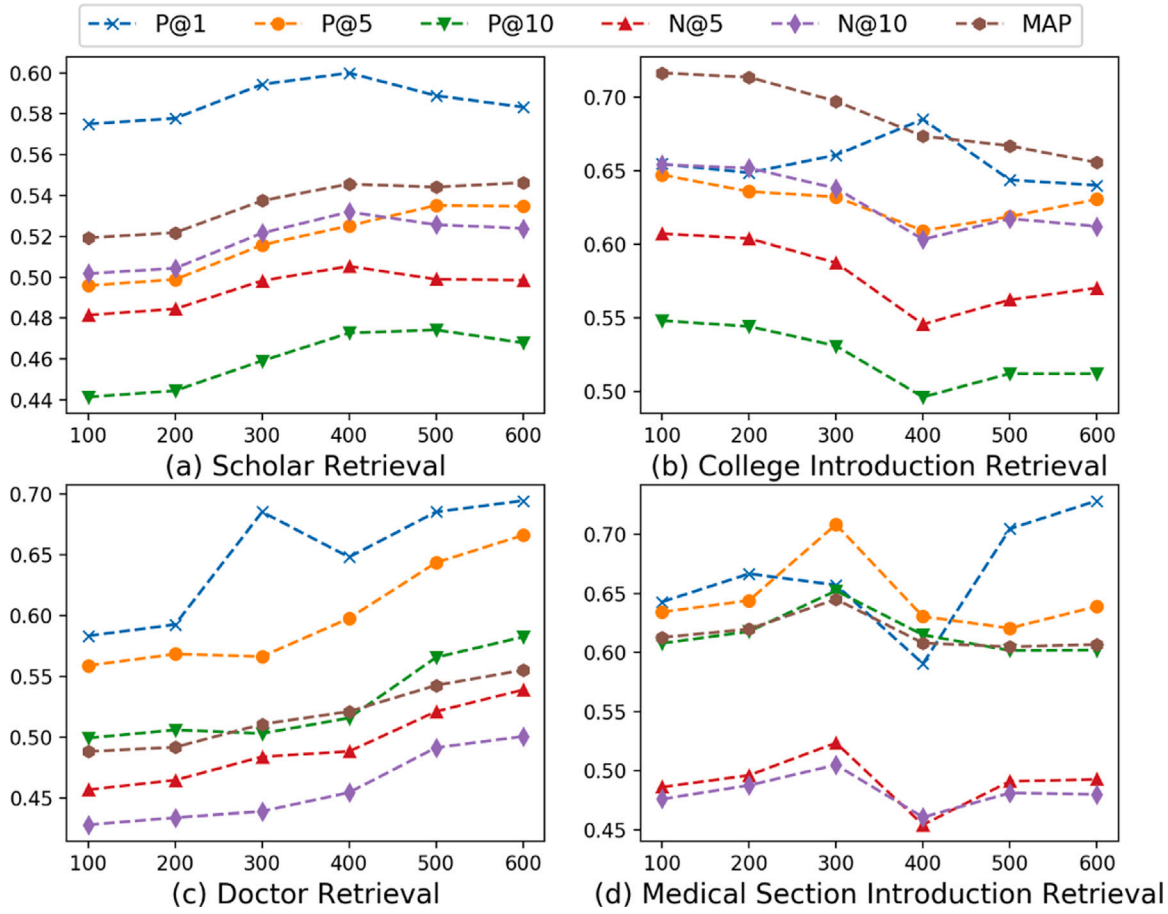



**Fig. 7.** LDA performance on different number of topics.

**Table 5**
Performance on scholar retrieval.

| Model | P@1 | P@5 | P@10 | R@5 | R@10 | N@5 | N@10 | MAP |
|---|---|---|---|---|---|---|---|---|
| VEM+TEM+FBS+$\alpha^*$ | **0.6667** | **0.6078** | **0.5095** | **0.4916** | **0.6880** | **0.5628** | **0.5927** | **0.6252** |
| VEM+TEM+FBS+$\alpha^0$ | 0.5833 | 0.5542 | 0.4557 | 0.3714 | 0.5847 | 0.5187 | 0.5335 | 0.5765 |
| VEM+TEM+$\alpha^*$ | 0.6333 | 0.5933 | 0.5016 | 0.3712 | 0.5340 | 0.5409 | 0.5630 | 0.6024 |
| VEM+LDA+$\alpha^*$ | 0.4833 | 0.4478 | 0.3957 | 0.2827 | 0.4652 | 0.3439 | 0.3811 | 0.4787 |
| VEM+LDA+$\alpha^0$ | 0.5000 | 0.4178 | 0.3892 | 0.2319 | 0.3934 | 0.3275 | 0.3723 | 0.4742 |
| VEM+BERT+$\alpha^*$ | 0.5333 | 0.5218 | 0.4540 | 0.3761 | 0.5632 | 0.4112 | 0.4589 | 0.5269 |
| VEM+BERT+$\alpha^0$ | 0.5167 | 0.4718 | 0.4347 | 0.3341 | 0.5734 | 0.3723 | 0.4270 | 0.5002 |
| VEM | 0.4500 | 0.4336 | 0.3737 | 0.2591 | 0.4058 | 0.3429 | 0.3687 | 0.4774 |
| VGG-16 | 0.2708 | 0.3099 | 0.2801 | 0.1880 | 0.3254 | 0.2544 | 0.2731 | 0.3623 |
| MobileNet (Howard et al., 2017) | 0.4500 | 0.4289 | 0.3761 | 0.2374 | 0.3819 | 0.3530 | 0.3983 | 0.4500 |
| $\epsilon$-ACSM (Amelio, 2019) | 0.2167 | 0.2304 | 0.2111 | 0.0987 | 0.1656 | 0.1481 | 0.1633 | 0.2850 |
| TEM+FBS | 0.6000 | 0.5077 | 0.4229 | 0.3361 | 0.5375 | 0.4846 | 0.4973 | 0.5184 |
| LDA | 0.3500 | 0.3182 | 0.2839 | 0.1839 | 0.3149 | 0.2307 | 0.2540 | 0.3544 |
| Q-BERT (Wang & Zhang, 2020) | 0.4833 | 0.3956 | 0.3357 | 0.2714 | 0.4170 | 0.3421 | 0.3500 | 0.4161 |
| HTML Similarity (Gowda & Mattmann, 2016) | 0.3333 | 0.3911 | 0.3952 | 0.1795 | 0.2335 | 0.2925 | 0.3216 | 0.4600 |
| URL2vec (Liang et al., 2019) | 0.2667 | 0.2742 | 0.2711 | 0.1625 | 0.2711 | 0.2353 | 0.2624 | 0.3782 |
| Kazemian's (Kazemian & Ahmed, 2015) | 0.1167 | 0.2724 | 0.2997 | 0.1111 | 0.2503 | 0.1734 | 0.2232 | 0.3327 |

#### *4.5.2. Performance on different tasks*

The overall performance on scholar retrieval, college introduction retrieval, doctor retrieval and medical section introduction retrieval is presented in Tables 5 to 8.

All seven models that consider both visual and textual features of webpages outperform others based on a single type of feature. It proves the backbone intuition that features in webpages' visual appearance and textual content are useful for webpage retrieval tasks. For instance, the effect of ignoring visual features can be seen from the comparison between VEM+TEM+FBS and TEM+FBS,

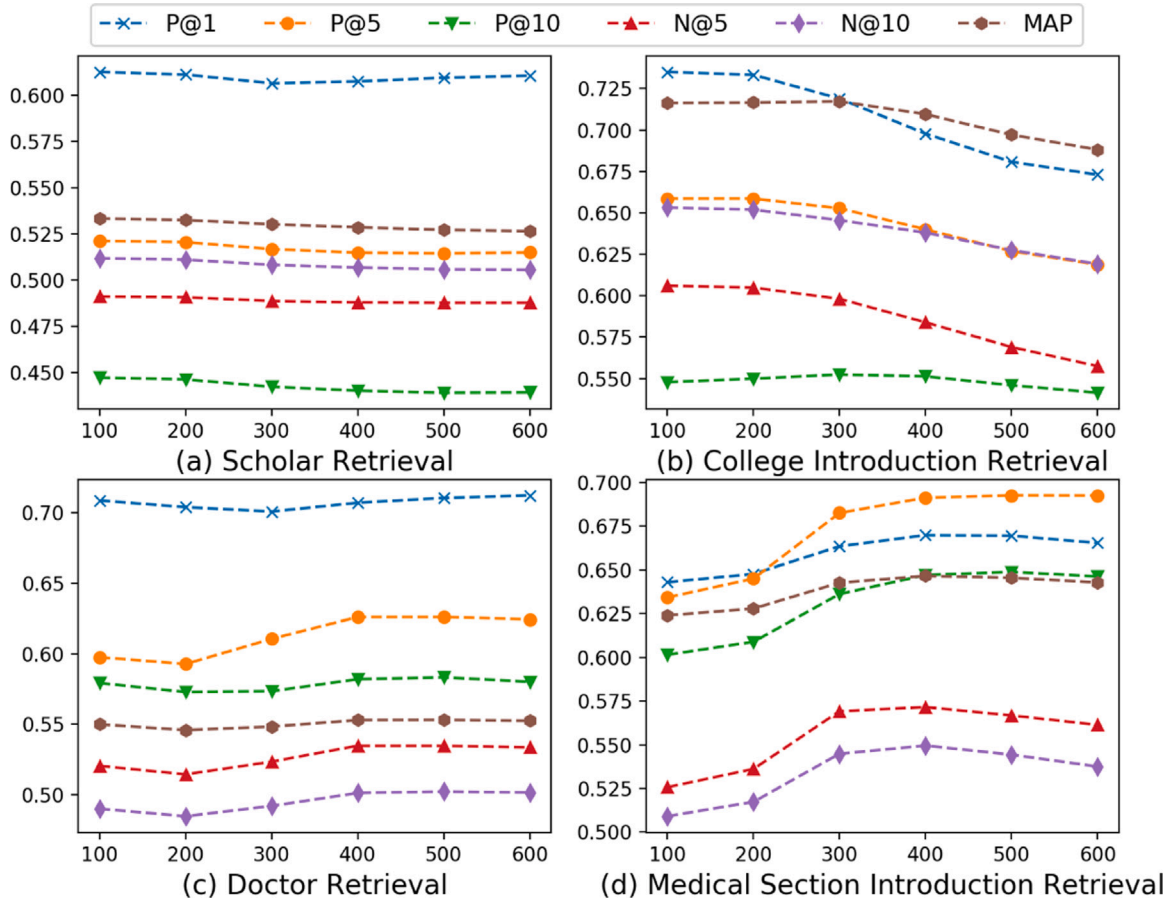



**Fig. 8.** TEM+FBS performance on different $\delta$.

VEM+LDA and LDA, VEM+BERT and BERT in Tables 5 to 8, which is consistent with the result of the sensitivity analysis for $\alpha$ that a small value of $\alpha$ degrades the performance. Theoretically, once webpages' visual features are ignored, the model depends on the textual content only. However, there are much more resources in other modalities to describe a webpage, which can be directly obtained from its visual appearance. Exactly, ignoring visual features means regarding webpages as plain texts only, and other informative features are missed, which leads to poor performance. Among all seven models, the proposed VEM+TEM+FBS+$\alpha^*$ achieves satisfactory performance under most circumstances, especially on expert retrieval tasks, which is fundamental to form a think tank.

As reported in Tables 5 to 8, baselines leveraging webpages' visual features fail to achieve satisfactory performance. Due to the structure of VGG-16 model, it can only process images sized $224 \times 224$, while the width and height of webpages' snapshots are usually larger than 500 pixels, and the aspect ratio is not always $1:1$. Over-resizing will blur the image, which influences the visual similarity estimation badly. Comparatively, MobileNet successfully makes a trade-off between effectiveness and efficiency, as it achieves much better performance with much less parameters than VGG-16. As for $\epsilon$-ACSM, the performance is not satisfactory on all tasks, which is probably because it estimates the similarity score directly using snapshots' original pixel values rather than learnt representations. In this way, the noise brought by snapshots' details and textures can adversely influence the similarity estimation, though patches in different images are matched approximately.

In comparisons with above baselines, the proposed VEM performs the best on all tasks, since the noise effect brought by snapshots' details and textures are largely alleviated, which is ignored in other baselines. In addition, the superiority of VEM is more obvious for expert retrieval task on Dataset 2, as shown in Table 7. It is probably caused by the fact that webpages in hospital websites usually contain more resources and have more complicated layouts compared to those in college websites. Thus, the advantage of a better visual feature extractor is much more obvious.

As for models based on textual features, the proposed TEM+FBS outperforms models based on LDA and BERT. Intuitively, the semi-structured texts do not have complete syntactic structures, like names and titles in webpages, which affects the effective representation of words from models on the basis of LDA and BERT. Besides, as a representative of LSTM-based models, BERT extracts textual features from the term grain, which is too precise for QBW. In other words, webpage pairs with different set of terms but sharing relevant topics should be retrieved by QBW models, while the estimated similarity score can be unexpectedly low

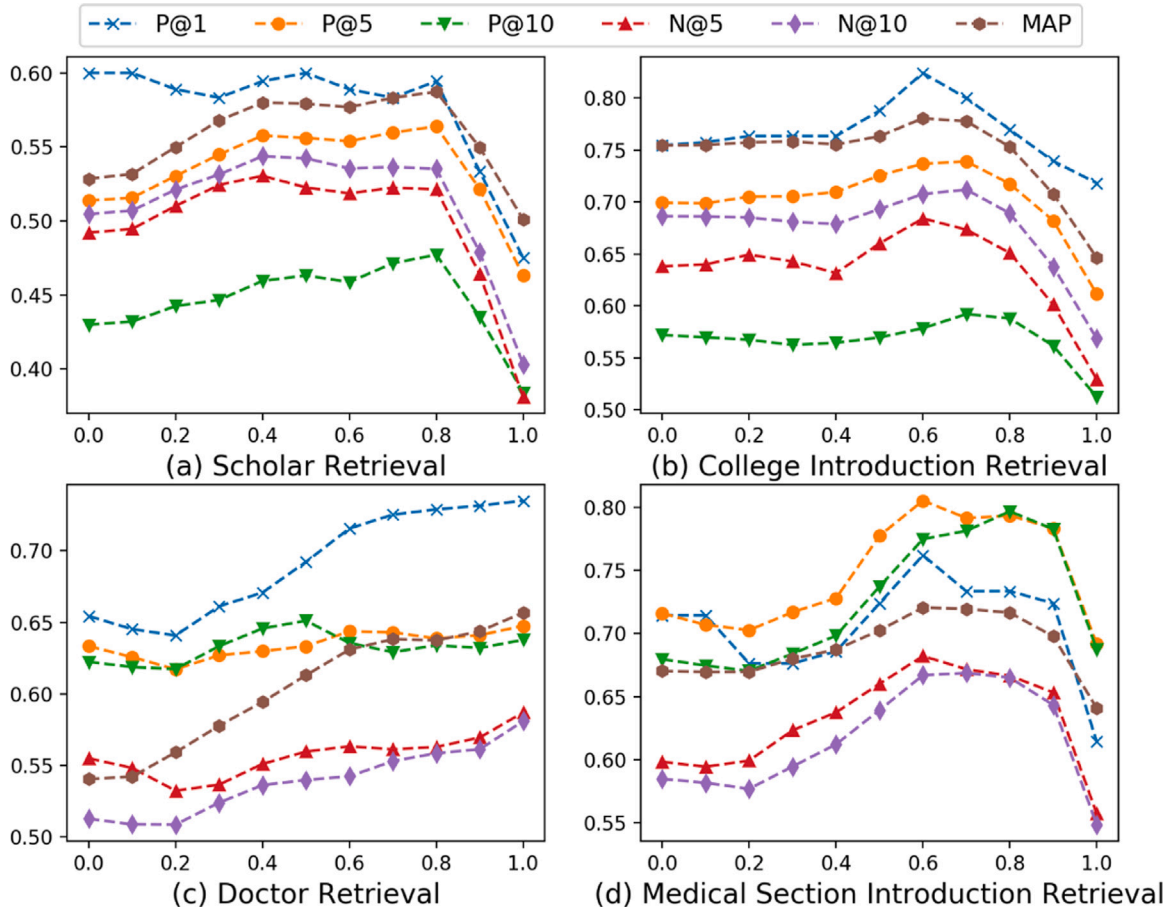



**Fig. 9.** VEM+TEM performance on different $\alpha$.

**Table 6**
Performance on college introduction retrieval.

| Model | P@1 | P@5 | P@10 | R@5 | R@10 | N@5 | N@10 | MAP |
|---|---|---|---|---|---|---|---|---|
| VEM+TEM+FBS+$\alpha^*$ | 0.8182 | 0.7306 | **0.6199** | 0.6266 | **0.8458** | 0.6454 | 0.7005 | 0.7738 |
| VEM+TEM+FBS+$\alpha^0$ | **0.8364** | **0.7393** | 0.5825 | **0.6267** | 0.8278 | **0.6883** | **0.7115** | **0.7869** |
| VEM+TEM+$\alpha^*$ | 0.6545 | 0.6197 | 0.5725 | 0.4560 | 0.7005 | 0.5536 | 0.6094 | 0.6851 |
| VEM+LDA+$\alpha^*$ | 0.6727 | 0.6526 | 0.5793 | 0.5110 | 0.7601 | 0.6099 | 0.6529 | 0.7063 |
| VEM+LDA+$\alpha^0$ | 0.6545 | 0.6796 | 0.5552 | 0.5413 | 0.7774 | 0.6117 | 0.6590 | 0.7069 |
| VEM+BERT+$\alpha^*$ | 0.7455 | 0.6442 | 0.5346 | 0.5019 | 0.7260 | 0.5993 | 0.6234 | 0.6665 |
| VEM+BERT+$\alpha^0$ | 0.7455 | 0.6492 | 0.5037 | 0.5153 | 0.7456 | 0.5799 | 0.6217 | 0.6627 |
| VEM | 0.7091 | 0.5978 | 0.4842 | 0.3926 | 0.6079 | 0.5195 | 0.5463 | 0.6063 |
| VGG-16 | 0.4909 | 0.4304 | 0.3519 | 0.3971 | 0.5517 | 0.3850 | 0.4264 | 0.5156 |
| MobileNet (Howard et al., 2017) | 0.5818 | 0.4739 | 0.4646 | 0.3555 | 0.5457 | 0.4374 | 0.4988 | 0.5419 |
| $\epsilon$-ACSM (Amelio, 2019) | 0.2364 | 0.3293 | 0.3004 | 0.2143 | 0.3061 | 0.2481 | 0.2902 | 0.3684 |
| TEM+FBS | 0.7455 | 0.7009 | 0.5787 | 0.5708 | 0.8213 | 0.6332 | 0.6872 | 0.7535 |
| LDA | 0.6364 | 0.5827 | 0.5054 | 0.5089 | 0.7477 | 0.5471 | 0.6072 | 0.6576 |
| Q-BERT (Wang & Zhang, 2020) | 0.7091 | 0.5773 | 0.4411 | 0.4693 | 0.6658 | 0.5320 | 0.5550 | 0.5905 |
| HTML Similarity (Gowda & Mattmann, 2016) | 0.2800 | 0.3494 | 0.3554 | 0.1699 | 0.2657 | 0.3100 | 0.3694 | 0.4291 |
| URL2Vec (Liang et al., 2019) | 0.5091 | 0.4034 | 0.3807 | 0.3413 | 0.4812 | 0.4180 | 0.4591 | 0.5196 |
| Kazemian's (Kazemian & Ahmed, 2015) | 0.2364 | 0.3384 | 0.3589 | 0.2047 | 0.3398 | 0.2779 | 0.3273 | 0.4173 |

if features from the term grain are utilized for estimation. In addition, with the combination of both textual and visual features, the disadvantage of LDA and BERT is mitigated to some extent, which implies the necessity and importance of both visual and textual features for QBW.

As for the baseline based on HTML documents, it achieves comparable performance with models only based on textual features on expert retrieval tasks, while its performance drops on institute introduction retrieval tasks obviously, especially on Dataset 2. As introduced in Section 2, HTML-based models transform HTML codes into a DOM tree for each webpage, and the similarity between

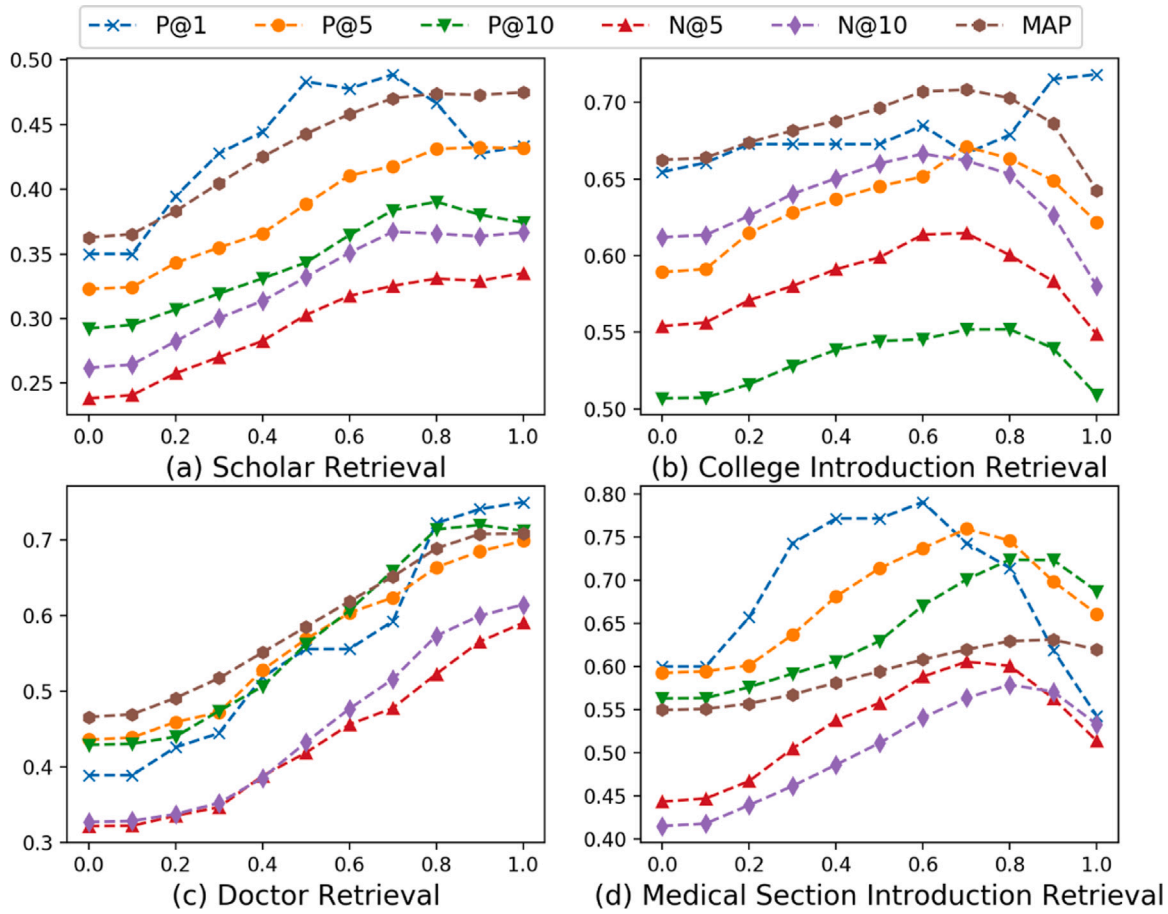



**Fig. 10.** VEM+LDA performance on different $\alpha$.

**Table 7**
Performance on doctor retrieval.

| Model | P@1 | P@5 | P@10 | R@5 | R@10 | N@5 | N@10 | MAP |
|---|---|---|---|---|---|---|---|---|
| VEM+TEM+FBS+$\alpha^*$ | **0.7778** | **0.7407** | 0.6818 | **0.2023** | **0.3853** | **0.6350** | **0.6148** | 0.7026 |
| VEM+TEM+FBS+$\alpha^0$ | 0.7222 | 0.6481 | 0.6266 | 0.1550 | 0.3010 | 0.5608 | 0.5484 | 0.6299 |
| VEM+TEM+$\alpha^*$ | 0.7222 | 0.6864 | 0.6783 | 0.1835 | 0.3502 | 0.5963 | 0.5856 | 0.6874 |
| VEM+LDA+$\alpha^*$ | 0.7222 | 0.7142 | **0.7354** | 0.1821 | 0.3842 | 0.5798 | 0.6050 | **0.7073** |
| VEM+LDA+$\alpha^0$ | 0.6111 | 0.6270 | 0.6805 | 0.1555 | 0.3422 | 0.4804 | 0.5286 | 0.6622 |
| VEM+BERT+$\alpha^*$ | 0.7222 | 0.7076 | 0.7106 | 0.1819 | 0.3714 | 0.5865 | 0.6038 | 0.6940 |
| VEM+BERT+$\alpha^0$ | 0.6667 | 0.7137 | 0.7167 | 0.1870 | 0.3754 | 0.5627 | 0.6012 | 0.6630 |
| VEM | 0.7378 | 0.6524 | 0.6453 | 0.1850 | 0.3718 | 0.6018 | 0.5992 | 0.6663 |
| VGG-16 | 0.4445 | 0.4672 | 0.5720 | 0.1075 | 0.2948 | 0.3381 | 0.4288 | 0.5908 |
| MobileNet (Howard et al., 2017) | 0.6667 | 0.6369 | 0.5778 | 0.1735 | 0.3006 | 0.5254 | 0.4953 | 0.6384 |
| $\epsilon$-ACSM (Amelio, 2019) | 0.1667 | 0.3253 | 0.3109 | 0.0604 | 0.1086 | 0.2268 | 0.2130 | 0.3508 |
| TEM+FBS | 0.7222 | 0.7061 | 0.6333 | 0.1867 | 0.3160 | 0.5950 | 0.5543 | 0.5854 |
| LDA | 0.3889 | 0.4278 | 0.4257 | 0.0995 | 0.1987 | 0.3206 | 0.3227 | 0.4541 |
| Q-BERT (Wang & Zhang, 2020) | 0.4444 | 0.4358 | 0.3945 | 0.1142 | 0.1992 | 0.3110 | 0.3151 | 0.4510 |
| HTML Similarity (Gowda & Mattmann, 2016) | 0.3889 | 0.2918 | 0.3286 | 0.0863 | 0.1846 | 0.2429 | 0.2643 | 0.4580 |
| URL2vec (Liang et al., 2019) | 0.3333 | 0.4342 | 0.4532 | 0.1035 | 0.2063 | 0.3161 | 0.3520 | 0.5207 |
| Kazemian's (Kazemian & Ahmed, 2015) | 0.1111 | 0.2546 | 0.3615 | 0.0442 | 0.1358 | 0.1651 | 0.2206 | 0.4393 |

DOM trees is taken as that between corresponding webpages. Considering that experts' webpages usually have explicit patterns to follow, the similarity between DOM trees can mirror that between the corresponding webpages to some extent. However, as the baseline is applied on general tasks like institute introduction retrieval, the assumption no longer exists, which results in poor performance.

Meanwhile, the performance of both URL-based methods is not satisfactory, which is probably resulted from two reasons. Firstly, there are a certain number of URLs can hardly provide useful information for webpage retrieval, like URL 2 in Table 3. If the URL

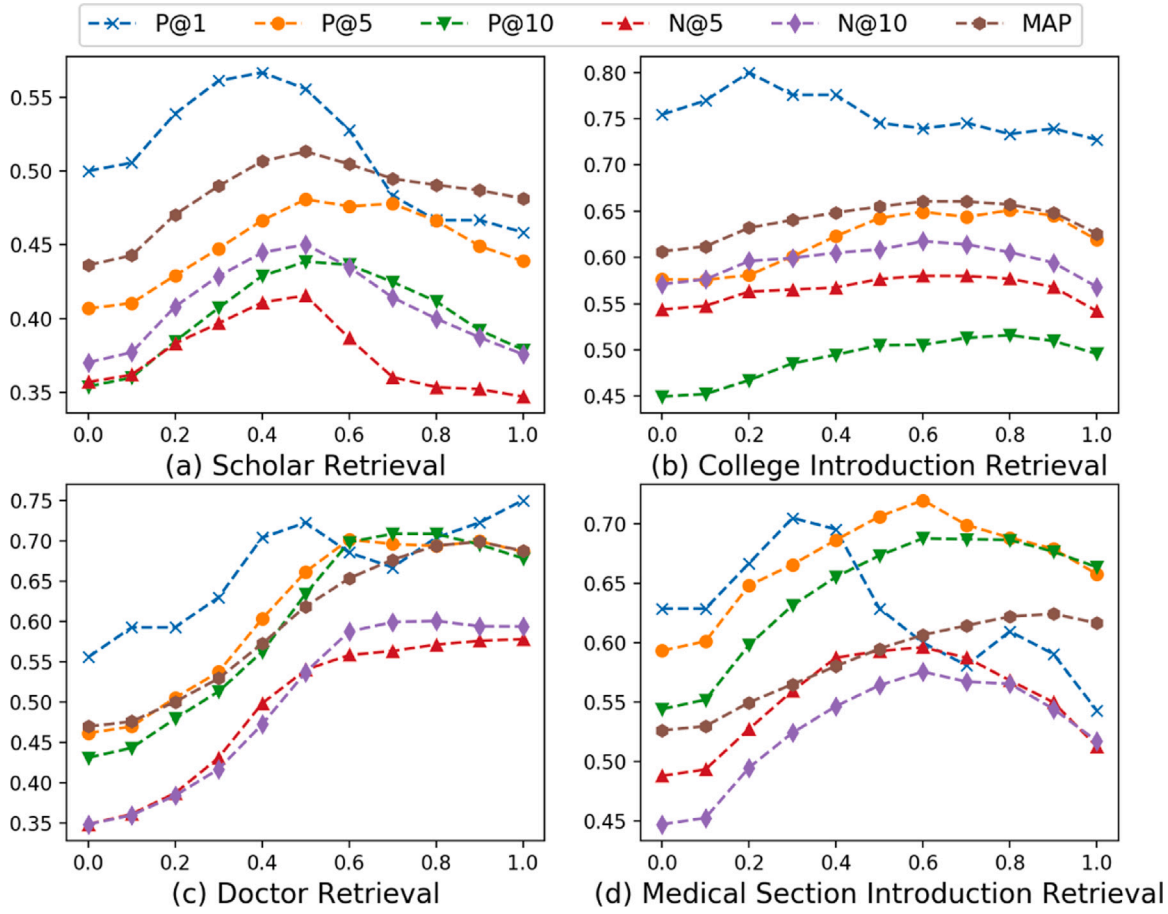



**Fig. 11.** VEM+BERT performance on different $\alpha$.

of either the exemplary or the candidate webpages is like URL 2 in Table 3, consisting of a string of meaningless characters, the retrieval performance will be adversely affected, no matter what feature extractor is applied. Secondly, similar to the problem faced by HTML-based methods, URLs in different websites are usually created according to their own coding rules. Thus, the similarity estimated via URL patterns is not equivalent to that between corresponding webpages. For instance, both webpages of URL 1 and 2 in Table 3 are scholar list in colleges' websites, and they are assumed to be similar in QBW, but URL 1 and 2 look completely different.

Moreover, URL2vec performs better compared to Kazemian's work that based on URLs' lexical features. On the one hand, neural networks in URL2vec encode rich information into the representation, especially the relation between tokens in a URL, while information conveyed by lexical features is usually limited. On the other hand, lexical features are designed heuristically for specific tasks like anti-phishing. Comparatively, URL2vec learns URLs' representation in a task-independent way like word2vec, i.e., predicting tokens in a URL given their previous tokens, so the learnt features can be extended to general tasks.

Besides, the performance of models with $\alpha^*$ and those with $\alpha^0$ proves the validity of Eq. (6). Specifically, models with $\alpha^*$ outperforms those with $\alpha^0$ by more than 5% in terms of most evaluation metrics on expert retrieval tasks. They also achieve comparable performance with models with $\alpha^0$ on institute introduction retrieval tasks. In practice, experts' webpages are usually more diverse in layout than institute introduction. For instance, the expert webpage in some websites comprises a large number of experts' photos and a small number of words like $p_0$ in Fig. 3, while those in other websites may contain fewer photos, even no photos at all, like Fig. 2(b) and (c). The discrimination of the visual similarity in the above three cases are quite different from each other, so weights of visual and textual similarity need to be assigned adaptively for different websites. On the contrary, institute introduction webpages are usually dominated by texts, probably together with a few images, so their visual appearance is as distinctive as each other in different websites. Thus, a pair of fixed weights can accommodate the characteristics of most websites on institute introduction retrieval tasks, and adaptive weighting is much less needed.

Note that, under most circumstances, the comparison result between any pair of models in terms of recall@k is consistent with that in terms of NDCG@k. Theoretically, recall is the fraction of target webpages that are retrieved. Due to the normalization of IDCG, NDCG evaluates models' performance from a similar perspective, which leads to consistent comparison results with those using recall.

**Table 8**
Performance on medical section introduction retrieval.

| Model | P@1 | P@5 | P@10 | R@5 | R@10 | N@5 | N@10 | MAP |
|---|---|---|---|---|---|---|---|---|
| VEM+TEM+FBS+$\alpha^*$ | 0.7143 | 0.7706 | **0.7861** | 0.1540 | 0.3142 | 0.6357 | 0.6420 | 0.6972 |
| VEM+TEM+FBS+$\alpha^0$ | **0.7714** | **0.8076** | 0.7849 | **0.1657** | **0.3209** | **0.6879** | **0.6751** | **0.7260** |
| VEM+TEM+$\alpha^*$ | 0.6857 | 0.7421 | 0.7398 | 0.1530 | 0.2828 | 0.6132 | 0.6074 | 0.6668 |
| VEM+LDA+$\alpha^*$ | 0.6571 | 0.7199 | 0.7087 | 0.1460 | 0.2824 | 0.5699 | 0.5649 | 0.6140 |
| VEM+LDA+$\alpha^0$ | 0.7143 | 0.7671 | 0.7077 | 0.1542 | 0.2816 | 0.6079 | 0.5699 | 0.6232 |
| VEM+BERT+$\alpha^*$ | 0.5714 | 0.7010 | 0.6808 | 0.1417 | 0.2769 | 0.5548 | 0.5544 | 0.6167 |
| VEM+BERT+$\alpha^0$ | 0.6000 | 0.7242 | 0.6923 | 0.1491 | 0.2763 | 0.5984 | 0.5789 | 0.6099 |
| VEM | 0.5143 | 0.6076 | 0.6038 | 0.1202 | 0.2379 | 0.4684 | 0.5645 | 0.5933 |
| VGG-16 | 0.4000 | 0.4681 | 0.5070 | 0.0961 | 0.2029 | 0.3517 | 0.3605 | 0.5241 |
| MobileNet (Howard et al., 2017) | 0.5143 | 0.5503 | 0.5126 | 0.1021 | 0.1808 | 0.3894 | 0.3776 | 0.5391 |
| $\epsilon$-ACSM (Amelio, 2019) | 0.3429 | 0.3252 | 0.3467 | 0.0530 | 0.1084 | 0.2541 | 0.2564 | 0.3844 |
| TEM+FBS | 0.7143 | 0.7433 | 0.6945 | 0.1529 | 0.2787 | 0.6099 | 0.5940 | 0.6721 |
| LDA | 0.6000 | 0.5869 | 0.5612 | 0.1196 | 0.2311 | 0.4328 | 0.4067 | 0.5464 |
| Q-BERT (Wang & Zhang, 2020) | 0.6286 | 0.5689 | 0.5200 | 0.1140 | 0.2101 | 0.4720 | 0.4313 | 0.5156 |
| HTML Similarity (Gowda & Mattmann, 2016) | 0.3143 | 0.1523 | 0.1682 | 0.0309 | 0.0684 | 0.1189 | 0.1296 | 0.3598 |
| URL2vec (Liang et al., 2019) | 0.2000 | 0.3029 | 0.4072 | 0.0511 | 0.1352 | 0.2482 | 0.2902 | 0.4975 |
| Kazemian's (Kazemian & Ahmed, 2015) | 0.2571 | 0.3123 | 0.3345 | 0.0565 | 0.1191 | 0.2520 | 0.2582 | 0.4385 |

Besides, almost all models' recall scores drop obviously on Dataset 2 compared to those on Dataset 1, while comparable NDCG and MAP scores are gained on the two datasets, as shown in Tables 5 to 8. It indicates that models' ranking ability is not degraded significantly when they are applied in different scenarios, and the recall score is probably degraded by the characteristics of Dataset 2. Particularly, there are more target webpages in Dataset 2 than in Dataset 1 as mentioned in Section 4.1, so the recall score on Dataset 2 has a smaller upper bound compared to that on Dataset 1, if a small number of webpages are retrieved. On the other hand, it is found that target webpages in hospitals' websites are more diverse in content than those in colleges' websites. Specifically, a certain number of sections in a hospital usually list doctors in their introduction page, while others may present doctors and the introduction on separate webpages, i.e., target webpages with heterogeneous content may be contained in one website at the meantime. As a result, target webpages sharing similar content with the given example are top ranked, which leads to a relatively promising score of precision@5 and precision@10, while those with different content can hardly be recalled in the top-ranked candidates, so the performance in terms of recall score is poor. In general, the decreased score of recall is resulted from the increasing number and variety of target webpages in Dataset 2.

#### *4.5.3. Error analysis*

In order to find the potential factors that degrade the proposed model's performance, a series of error analyses were conducted on two representative cases, focusing on institute introduction and expert retrieval respectively.

**Case 1: Institute Introduction Retrieval**

It was found that VEM+TEM+FBS+$\alpha^*$ performs poorly when searching for college introduction from website $w_c$[1] using $q_{c1}$[2] as the exemplary webpage. To be specific, the average precision $AP = 0.13$.

The sensitivity of model's overall performance to different weights of visual and textual similarity was firstly analyzed, to check the effectiveness of the proposed weighting scheme. As shown in Fig. 12, the value of average precision hardly changes with different values of $\alpha$, which means that none of textual or visual similarity is significantly distinctive for the case. The current value of $\alpha$ is given as 0.6 by the weighting scheme, which balances the effect of both visual and textual similarity. Thus, the weighting scheme works fine, and the problem should be within either VEM or TEM.

Then, the value of $\alpha$ was set to be 1.0 to analyze the performance merely based on visual features, and the top 5 ranked webpages are listed in Table 9. It can be seen that even though the top ranked webpages are irrelevant, they do look similar to $q_{c1}$. Particularly, they are all consisted of large paragraphs together with one or two images. It implies that VEM can help to retrieve webpages sharing similar appearance with the given example, and it is the textual features that are needed for further filtering.

Accordingly, model's performance based on textual features was evaluated, however, it is not satisfactory, since a large number of irrelevant webpages are top ranked, as shown in Table 10. Hence, the output of each step in TEM was examined for the webpages, and some proper nouns in $q_{c1}$ were found incorrectly recognized. Particularly, the POS tagger in HanLP package is applied in TEM for term recognition, and the proper nouns are then replaced according to the corresponding category label, as indicated in Section 3.3.1, yet the range of category labels is limited. For instance, there are a large number of proper nouns in business domain used in $q_{c1}$ as well as the false positive webpages in Table 10, and such terms are unfortunately out of the category labels of HanLP. Note that, false positive webpages refer to irrelevant but top ranked webpages, i.e., incorrectly retrieved ones. On the contrary, most proper nouns in the target webpage of $w_c$ are in mathematics domain which can be correctly recognized by the POS tagger and replaced according to pre-defined rules. The topic distribution of each webpage mirrors such mistake. Specifically, it was found that the topic

[1] http://maths.whu.edu.cn/

[2] http://bs.bnu.edu.cn/xygk/xyjj/index.html

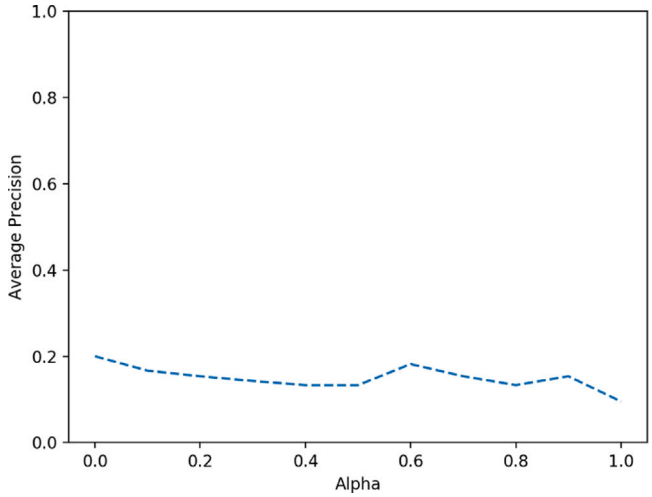



**Fig. 12.** AP on different $\alpha$ in Case 1.

**Table 9**
Top 5 ranked webpages based on visual features in case 1.

| Rank | URL | Label |
|---|---|---|
| 1 | http://maths.whu.edu.cn/info/1203/6507.htm | 0 |
| 2 | http://maths.whu.edu.cn/info/1203/6508.htm | 0 |
| 3 | http://maths.whu.edu.cn/info/1202/6663.htm | 0 |
| 4 | http://maths.whu.edu.cn/info/1323/13941.htm | 0 |
| 5 | http://maths.whu.edu.cn/info/1203/10605.htm | 0 |

**Table 10**
Top 5 ranked webpages based on textual features in case 1.

| Rank | URL | Label |
|---|---|---|
| 1 | http://maths.whu.edu.cn/kxyj/6.htm | 0 |
| 2 | http://maths.whu.edu.cn/kxyj/5.htm | 0 |
| 3 | http://maths.whu.edu.cn/kxyj.htm | 0 |
| 4 | http://maths.whu.edu.cn/kxyj/3.htm | 0 |
| 5 | http://maths.whu.edu.cn/kxyj/1.htm | 0 |

distribution of $q_{c1}$ and false positive webpages overlaps in topics consisted of business terms, while the target webpage most likely falls in topics comprising general academic terms. Therefore, the textual similarity between $q_{c1}$ and false positive webpages is much higher estimated than that between $q_{c1}$ and the target one, which makes it harder to be identified.

To validate the inference, the training dataset of the POS tagger in HanLP was supplemented with a set of business terms, such that proper nouns in $q_{c1}$ can be correctly identified and replaced. In this way, the target webpage is first ranked, and the average precision is enhanced from 0.13 to 1.00. Moreover, $q_{c2}$[3] was also used as the exemplary webpage, which is the introduction of a mathematics school, to search for introduction page from $w_c$. Since most proper nouns in $q_{c2}$ are in mathematics domain and can be correctly labeled and replaced like those in the target webpage in $w_c$, a satisfactory result is achieved ($AP = 1.00$).

The proposed model faces similar problem on Dataset 2 as well, because there are a large number of technical terms in medical scenario that can hardly be recognized by a general POS tagger. This case implies that the performance of the proposed model can be affected by the result of components to some extent. In the future, such problem is hoped to be alleviated by replacing existing components with targeted ones aimed for think tank construction. For instance, a POS tagger trained with proper nouns from various subjects can be used to fix the problem in Case 1.

**Case 2: Expert Retrieval**

When searching for doctor lists from a hospital website $w_h$,[4] the performance drops if $q_h$[5] is used as the exemplary webpage. Specifically, the average precision $AP = 0.33$, which is much lower than the MAP.

[3] https://www.math.pku.edu.cn/xygk/xyjj/index.htm
[4] https://ss.bjmu.edu.cn/Html/
[5] https://www.anzhen.org/Departments/Main/SearchIndex?siteId=224

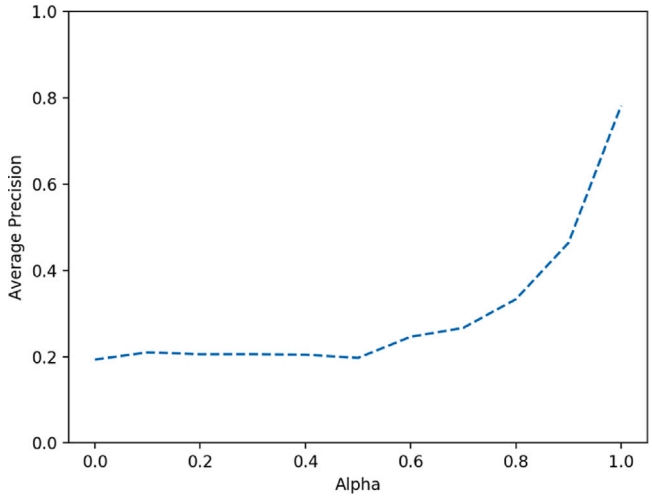



**Fig. 13.** AP on different $\alpha$ in Case 2.

**Table 11**
Top 5 ranked webpages based on textual features in case 2.

| Rank | URL | Label |
|---|---|---|
| 1 | https://ss.bjmu.edu.cn/Hospitals/Diseases/ListD | 0 |
| 2 | https://ss.bjmu.edu.cn/Html/News/Articles/101072.html | 0 |
| 3 | https://ss.bjmu.edu.cn/Html/News/Articles/101096.html | 0 |
| 4 | https://ss.bjmu.edu.cn/Html/News/Articles/600.html | 0 |
| 5 | https://ss.bjmu.edu.cn/Html/Departments/Main/Index_717.html | 1 |

First of all, the retrieval performance was evaluated in terms of average precision based on different values of $\alpha$. As shown in Fig. 13, the overall performance improves obviously as the value of $\alpha$ increases, which means that webpages' visual features are more distinctive in this case. The proposed weighting scheme calculates the value of $\alpha$ as 0.8. It makes visual features contribute more to the overall similarity estimation, which is consistent with the implication in Fig. 13. Besides, Fig. 13 also proves the validity of VEM in this case, since the value of average precision is promising when $\alpha = 1.0$. Therefore, the problem should not be in the component of weighting scheme or VEM.

Accordingly, only textual features were applied to search doctor lists in $w_h$ using $q_h$, and the performance is not satisfactory. As shown in Table 11, none of doctor lists in $w_h$ is top ranked. To find what reduced the estimated textual similarity between $q_h$ and doctor lists in $w_h$, their textual contents were compared, and the non-structured contents were found different. Note that, $q_h$ can be regarded as one typical pattern in different institutes' websites, whose doctor lists are contained in their introduction page. However, the doctor lists and institute introduction in $w_h$ are placed in separate webpages. In other words, the non-structured content of $q_h$ contains the institute introduction, while that of doctor lists in $w_h$ only contain doctors' expertise. Thus, their textual similarity is lower estimated, which imposes difficulties on recalling doctor lists from $w_h$ using $q_h$. To prove the assumption, textual features of non-structured content in all webpages were dropped out, based on which doctor lists in $w_h$ were searched using $q_h$ with $\alpha = 0.8$. It turned out that the performance is obviously improved ($AP = 0.70$).

The problem in Case 2 is partially caused by the proposed FBS measure, since features are randomly selected for similarity estimation without filtering. However, a textual similarity metric based on feature bootstrapping is necessary, because it helps to filter out irrelevant webpages that share a small proportion of features extremely similar to those of the given webpage. Practically, the effectiveness of FBS can be seen from Tables 5 to 8 by comparing VEM+TEM+FBS+$\alpha^*$ and VEM+TEM+$\alpha^*$. Therefore, the problem can be alleviated by proposing an interactive textual similarity measure, which is hoped to select features intelligently rather than randomly by leveraging supervision from users. For instance, in Case 2, a user will be allowed to select regions from the exemplary webpage that matter to his/her task, and textual features from the selected regions will be assigned with higher weights for similarity estimation, so that the textual feature of institute introduction in $q_h$ can be eluded.

## 5. Implications

This study is intended to explore the potential value of QBW in collecting information for think tank construction. Useful insights can also be obtained for representation learning methods of webpage, image and text. To summarize, both practical and theoretical implications of this study are elaborated in this section.

### 5.1. Practical implications

Remarks that are of practical use can be derived from this study for researchers who need to build a think tank. Given the massive amount and variety of webpages, it is demanding for researchers to manually collect sufficient experts' and institutes' information. Even if the think tank is constructed, its quality is questionable, because manual filtering is inevitably subjective. The proposed model is hoped to alleviate such problems and arouse the attention of academic community to the potential value of QBW in think tank construction. Specifically, required webpages of experts and institutes can be returned from target websites automatically, according to their similarity with the examples provided by researchers. Since manual filtering and high-quality queries are no longer needed, the efficiency and objectivity of think tank construction can be largely improved.

Besides, readers of this study also include those whose work is associated with similarity or distance between webpages, including programmers in charge of webpage archive maintenance or anti-phishing, online shopping service providers who need to recommend similar products to users, headhunters searching for qualified talents, etc. Considering that most components within the proposed model are replaceable, it can be easily transferred to various scenarios.

### 5.2. Theoretical implications

This study sheds some light on the literature of webpage representation learning. Specifically, the representation method for webpages needs to adapt to the characteristics of different websites, so that the learnt feature vector can be applied for cross-website tasks, e.g., webpage classification, clustering, retrieval. Comparatively, features extracted from webpages' URLs and HTML documents in existing studies are hardly scalable, due to the inconsistent version and coding habits. Inspired by human beings' searching behaviors, this study provides a general way to represent a webpage, leveraging its appearance and content.

Accordingly, theoretical implications can also be derived for representation learning and similarity estimation of text and images. For QBE tasks, the representation should not be over-precise, otherwise, the similarity between relevant samples can be lower estimated, which makes them hard to be recalled. Particularly, the satisfactory result of the proposed model implies that topic distribution and high-layer features in CNNs can represent webpages' textual content and snapshots respectively in an appropriately coarse grain, which is worth further studied. Besides, it is proved that feature bootstrapping can improve the robustness of the estimated similarity. Its effect can be much more promising, if golden label or users' supervision is available, according to the error analysis.

## 6. Conclusion

Due to the dramatic growth in the number and variety of webpages, there is a great need for an automatic webpage retrieval model for think tank construction as well as other general tasks. The principal insight gained from this work is the value of combining webpages' visual and textual features for retrieval tasks. Comparing to existing methods, the proposed visual feature extraction module can obtain high-level features from webpages' snapshots which is much less affected by the noise effect brought by images. Meanwhile, a textual feature extraction module is proposed to process webpages' textual content. Features from both term and topic level are utilized to find an appropriately coarse grain for QBW. In addition, a series of metrics are proposed to measure the similarity between webpages. Extensive experiments prove that the proposed model is effective and can accommodate refinements.

In the future, the performance and generality of the proposed model will be evaluated in more real-world applications, like similar products retrieval, policy document retrieval, etc., so that the target users can be enlarged. Secondly, for a given example webpage, critical keywords are hoped to be extracted from its textual content and returned to users. Users will be allowed to revise them and form a query, and similar webpages will be subsequently retrieved according to the user-revised query. In this way, both the interactivity and the time efficiency of textual feature extraction module will be improved, compared to modeling the entire textual content. Thirdly, the time cost of the proposed model can be further reduced by optimizing crawling programs to obtain webpages' resources which is the most time-consuming process for now.

## CRediT authorship contribution statement

**Qian Geng:** Conceptualization, Supervision, Resources. **Ziang Chuai:** Data curation, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Jian Jin:** Conceptualization, Funding acquisition, Formal analysis, Project administration, Writing – review & editing.

## Acknowledgments

This work was supported by a National Social Science Foundation of China (Grant. 19ATQ005) and a National Natural Science Foundation of China (NSFC 71701019/G0114).

## References


Amato, G., Carrara, F., Falchi, F., Gennaro, C., & Vadicamo, L. (2020). Large-scale instance-level image retrieval. *Information Processing and Management*, *57*(6), Article 102100.

Amelio, A. (2019). A new axiomatic methodology for the image similarity. *Applied Soft Computing*, *81*, Article 105474.

Arshed, N. (2017). The origins of policy ideas: The importance of think tanks in the enterprise policy process in the UK. *Journal of Business Research*, *71*, 74–83.

Bohunsky, P., & Gatterbauer, W. (2010). Visual structure-based web page clustering and retrieval. In *Proceedings of the 19th international conference on world wide web* (pp. 1067–1068). Raleigh, North Carolina, USA.

Bozkır, A. S., & Sezer, E. A. (2014). SimiLay: A developing web page layout based visual similarity search engine. In P. Petra (Ed.), *Machine learning and data mining in pattern recognition* (pp. 457–470). Cham.

Bozkir, A. S., & Sezer, E. A. (2018). Layout-based computation of web page similarity ranks. *International Journal of Human-Computer Studies*, *110*, 95–114.

Chen, H.-H., Treeratpituk, P., Mitra, P., & Giles, C. L. (2013). CSSeer: An expert recommendation system based on CiteseerX. In *Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries* (pp. 381–382). Indianapolis, Indiana, USA.

Chen, C., Zhang, Y.-L., Qiu, M., Wu, B., Wang, L., Li, L., et al. (2020). Automatic knowledge fusion in transferrable networks for semantic text matching. In *Companion proceedings of the web conference 2020* (pp. 73–74). Taipei, Taiwan.

Dargahi Nobari, A., Neshati, M., & Sotudeh Gharebagh, S. (2020). Quality-aware skill translation models for expert finding on stack overflow. *Information Systems*, *87*, Article 101413.

Dehghan, M., Abin, A. A., & Neshati, M. (2020). An improvement in the quality of expert finding in community question answering networks. *Decision Support Systems*, *139*, Article 113425.

Dehghan, M., Biabani, M., & Abin, A. A. (2019). Temporal expert profiling: With an application to t-shaped expert finding. *Information Processing and Management*, *56*(3), 1067–1079.

Dourado, I. C., Galante, R., Gonçalves, M. A., & da Silva Torres, R. (2019). Bag of textual graphs (BoTG): A general graph-based text representation model. *Journal of the Association for Information Science and Technology*, *70*(8), 817–829.

Fan, Y., Guo, J., Lan, Y., Xu, J., Pang, L., & Cheng, X. (2017). Learning visual features from snapshots for web search. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 247–256). Singapore, Singapore.

Faraday, P. (2000). Visually critiquing web pages. In *Multimedia '99* (pp. 155–166). Vienna.

Goodolf, D. M., & Godfrey, N. (2020). A think tank in action: Building new knowledge about professional identity in nursing. *Journal of Professional Nursing*.

Gowda, T., & Mattmann, C. A. (2016). Clustering web pages based on structure and style similarity (application paper). In *2016 IEEE 17th international conference on information reuse and integration* (pp. 175–180). Hanoi, Viet Nam.

Ha-Thuc, V., Yan, Y., Wu, X., Dialani, V., Gupta, A., & Sinha, S. (2017). From query-by-keyword to query-by-example: LinkedIn talent search approach. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 1737–1745). Singapore, Singapore.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916.

Hernando, M. G., Pautz, H., & Stone, D. (2018). Think tanks in 'hard times' – the global financial crisis and economic advice. *Policy and Society*, *37*(2), 125–139.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. CoRR arXiv:1704.04861.

Jaeyoung, K., Janghyeok, Y., Eunjeong, P., & Sungchul, C. (2020). Patent document clustering with deep embeddings. *Scientometrics*, *123*, 563–577.

Jeh, G., & Widom, J. (2002). SimRank: A measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 538–543). Edmonton, Alberta, Canada.

Jiang, J.-Y., Zhang, M., Li, C., Bendersky, M., Golbandi, N., & Najork, M. (2019). Semantic text matching for long-form documents. In *The world wide web conference* (pp. 795–806). San Francisco, CA, USA.

Jun, S., Park, S.-S., & Jang, D.-S. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, *41*(7), 3204–3212.

Kazemian, H., & Ahmed, S. (2015). Comparisons of machine learning techniques for detecting malicious webpages. *Expert Systems with Applications*, *42*(3), 1166–1177.

Kim, K. (2018). An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis. *Expert Systems with Applications*, *109*, 49–65.

Lakshmi, R., & Baskar, S. (2019). Novel term weighting schemes for document representation based on ranking of terms and fuzzy logic with semantic relationship of terms. *Expert Systems with Applications*, *137*, 493–503.

Law, M. T., Thome, N., Gançarski, S., & Cord, M. (2012). Structural and visual comparisons for web page archiving. In *Proceedings of the 2012 ACM symposium on document engineering* (pp. 117–120). Paris, France.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. in *Proceedings of the 31st international conference on machine learning* (pp. 1188–1196). Beijing, China.

Li, J., Dou, Z., Zhu, Y., Zuo, X., & Wen, J.-R. (2019). Deep cross-platform product matching in e-commerce. *Information Retrieval Journal*, *23*(2), 136–158.

Li, Y., Xu, S., Luo, X., & Lin, S. (2014). A new algorithm for product image search based on salient edge characterization. *Journal of the Association for Information Science and Technology*, *65*(12), 2534–2551.

Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems*, *94*, 27–39.

Li, X., Yang, J., & Ma, J. (2021). Recent developments of content-based image retrieval (CBIR). *Neurocomputing*.

Liang, S. (2019). Unsupervised semantic generative adversarial networks for expert retrieval. In *The world wide web conference* (pp. 1039–1050). San Francisco, CA, USA.

Liang, S., & de Rijke, M. (2016). Formal language models for finding groups of experts. *Information Processing and Management*, *52*(4), 529–549.

Liang, Y., Kang, J., Yu, Z., Guo, B., Zheng, X., & He, S. (2019). Leverage temporal convolutional network for the representation learning of URLs. In *2019 IEEE international conference on intelligence and security informatics* (pp. 74–79).

Lin, Z., Lyu, M. R., & King, I. (2006). pagesim: a novel link-based measure of web page similarity for the world wide web. In *Proceedings of the 15th international conference on world wide web* (pp. 1019–1020). Edinburgh, Scotland.

Lin, Z., Lyu, M. R., & King, I. (2009). MatchSim: A novel neighbor-based similarity measure with maximum neighborhood matching. In *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 1613–1616). Hong Kong, China.

Lopez-Otero, P., Parapar, J., & Barreiro, A. (2019). Efficient query-by-example spoken document retrieval combining phone multigram representation and dynamic time warping. *Information Processing and Management*, *56*(1), 43–60.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

McGann, J. G. (2020). *2019 global go to think tank index report: Tech. rep. 17*, TTCSP Global Go To Think Tank Index Reports.

Nguyen, L. D., Le, D.-N., & Vinh, L. T. (2014). Detecting phishing web pages based on DOM-tree structure and graph matching algorithm. In *Proceedings of the fifth symposium on information and communication technology* (pp. 280–285). Hanoi, Viet Nam.

Pinheiro, R. H., Cavalcanti, G. D., & Tsang, I. R. (2017). Combining dissimilarity spaces for text categorization. *Information Sciences*, *406–407*, 87–101.
Planells-Artigot, E., Ortigosa-Blanch, A., & Martí-Sánchez, M. (2021). Bridging fields: A comparative study of the presence of think tanks. *Technological Forecasting and Social Change*, *162*, Article 120377.
Rinaldi, A. M., Russo, C., & Tommasino, C. (2020). A semantic approach for document classification using deep neural networks and multimedia knowledge graph. *Expert Systems with Applications*, *169*, Article 114320.
Roostaee, M., Sadreddini, M. H., & Fakhrahmad, S. M. (2020). An effective approach to candidate retrieval for cross-language plagiarism detection: A fusion of conceptual and keyword-based schemes. *Information Processing and Management*, *57*(2), Article 102150.
Rostami, P., & Neshati, M. (2019). T-shaped grouping: Expert finding models to agile software teams retrieval. *Expert Systems with Applications*, *118*, 231–245.
Ruser, A. (2018). What to think about think tanks: Towards a conceptual framework of strategic think tank behaviour. *International Journal of Politics, Culture, and Society*, *31*(2), 179–192.
Sarwar, S. M., & Allan, J. (2020). Query by example for cross-lingual event retrieval. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 1601–1604). China: Virtual Event.
Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 640–651.
Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 4080–4090). Long Beach, California, USA.
Song, W., Liang, J. Z., & Park, S. C. (2014). Fuzzy control GA with a novel hybrid semantic similarity strategy for text clustering. *Information Sciences*, *273*, 156–170.
Takama, Y., & Mitsuhashi, N. (2005). Visual similarity comparison for web page retrieval. In *Proceedings of the 2005 IEEE/WIC/ACM international conference on web intelligence* (pp. 301–304). USA.
van den Akker, B., Markov, I., & de Rijke, M. (2019). ViTOR: Learning to rank webpages based on visual features. In *The world wide web conference* (pp. 3279–3285). San Francisco, CA, USA.
Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In *Proceedings of the 30th international conference on neural information processing systems* (pp. 3637–3645). Barcelona, Spain.
Wang, C., & Zhang, X. (2020). Q-BERT: A BERT-based framework for computing SPARQL similarity in natural language. In *Companion proceedings of the web conference 2020* (pp. 65–66). Taipei, Taiwan.
Weng, L., Li, Z., Cai, R., Zhang, Y., Zhou, Y., Yang, L. T., et al. (2011). Query by document via a decomposition-based two-level retrieval approach. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 505–514). Beijing, China.
Wu, D., Fan, S., & Yuan, F. (2021). Research on pathways of expert finding on academic social networking sites. *Information Processing and Management*, *58*(2), Article 102475.
Wu, C., Kanoulas, E., & de Rijke, M. (2020). Learning entity-centric document representations using an entity facet topic model. *Information Processing and Management*, *57*(3), Article 102216.
Wu, Z., Zhu, H., Li, G., Cui, Z., Huang, H., Li, J., et al. (2017). An efficient wikipedia semantic matching approach to text document classification. *Information Sciences*, *393*, 15–28.
Xu, C., Lin, Z., Wu, S., & Wang, H. (2019). Multi-level matching networks for text matching. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 949–952). Paris, France.
Yan, D., Li, K., Gu, S., & Yang, L. (2020). Network-based bag-of-words model for text classification. *IEEE Access*, *8*, 82641–82652.
Yan, Y., Liu, T., Shi, J., Wang, Q., & Guo, L. (2017). Flexible expert finding on the Web via semantic hypergraph learning and affinity propagation model. In *2017 IEEE 29th international conference on tools with artificial intelligence* (pp. 1204–1209). Boston, MA, USA.
Yu, W., Yang, K., Yao, H., Sun, X., & Xu, P. (2017). Exploiting the complementary strengths of multi-layer CNN features for image retrieval. *Neurocomputing*, *237*, 235–241.
Zhang, W., Li, Y., & Wang, S. (2019). Learning document representation via topic-enhanced LSTM model. *Knowledge-Based Systems*, *174*, 194–204.
Zhao, R., & Mao, K. (2018). Fuzzy bag-of-words model for document representation. *IEEE Transactions on Fuzzy Systems*, *26*(2), 794–804.
Zhao, Z., Zhu, H., Xue, Z., Liu, Z., Tian, J., Chua, M. C. H., et al. (2019). An image-text consistency driven multimodal sentiment analysis approach for social media. *Information Processing and Management*, *56*(6), Article 102097.