A Deep Relevance Model for Zero-Shot Document Filtering

Chenliang Li[1], **Wei Zhou**[2], **Feng Ji**[2], **Yu Duan**[1], **Haiqing Chen**[2]
[1]School of Cyber Science and Engineering, Wuhan University, China
{cllee,duanyu}@whu.edu.cn
[2]Alibaba Group, Hangzhou, China
{fayi.zw,zhongxiu.jf,haiqing.chenhq}@alibaba-inc.com

Abstract

In the era of big data, focused analysis for diverse topics with a short response time becomes an urgent demand. As a fundamental task, information filtering therefore becomes a critical necessity. In this paper, we propose a novel deep relevance model for zero-shot document filtering, named DAZER. DAZER estimates the relevance between a document and a category by taking a small set of seed words relevant to the category. With pre-trained word embeddings from a large external corpus, DAZER is devised to extract the relevance signals by modeling the hidden feature interactions in the word embedding space. The relevance signals are extracted through a gated convolutional process. The gate mechanism controls which convolution filters output the relevance signals in a category dependent manner. Experiments on two document collections of two different tasks (*i.e.,* topic categorization and sentiment analysis) demonstrate that DAZER significantly outperforms the existing alternative solutions, including the state-of-the-art deep relevance ranking models.

1 Introduction

Filtering irrelevant information and organizing relevant information into meaningful topical categories is indispensable and ubiquitous. For example, a data analyst tracking an emerging event would like to retrieve the documents relevant to a specific topic (category) from a large document collection in a short response time. In the era of big data, the potentially possible categories covered by documents would be limitless. It is unrealistic to manually identify a lot of positive examples for each possible category. However, new information needs indeed emerge everywhere in many real-world scenarios. Recent studies on dataless text classification show promising results on reducing labeling effort (Liu et al., 2004; Druck et al., 2008; Chang et al., 2008; Song and Roth, 2014; Hingmire et al., 2013; Hingmire and Chakraborti, 2014; Chen et al., 2015; Li et al., 2016). Without any labeled document, a dataless classifier performs text classification by using a small set of relevant words for each category (called "seed words"). However, existing dataless classifiers do not consider *document filtering*. We need to provide the seed words for each category covered by the document collection, which is often infeasible in the real world.

To this end, we are particularly interested in the task of zero-shot document filtering. Here, *zero-shot* means that the instances of the targeted categories are unseen during the training phase. To facilitate zero-shot filtering, we take a small set of seed words to represent a category of interest. This is extremely useful when the information need (*i.e.,* the categories of interest) is dynamic and the text collection is large and temporally updated (*e.g.,* the possible categories are hard to know). Specifically, we propose a novel **d**eep relev**a**nce model for **z**ero-shot document filt**er**ing, named DAZER. In DAZER, we use the word embeddings learnt from an external large text corpus to represent each word. A category can then be well represented also in the embedding space (called *category embedding*) through some composition with the word embeddings of the provided seed words. Given a small number of seed words provided for a category as input, DAZER is devised to produce a score indicating the relevance between a document and the category. It is intuitive to connect zero-shot document filtering

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 2300–2310
Melbourne, Australia, July 15 - 20, 2018.

with the task of ad-hoc retrieval. Indeed, by treating the seed words of each category as a query, the zero-shot document filtering is equivalent to ranking documents based on their relevance to the query. The relevance ranking is a core task in information retrieval, and has been studied for many years. Although they share the same formulation, these two tasks diverge fundamentally. For ad-hoc retrieval, a user constructs a query with a specific information need. The relevant documents are assumed to contain these query words. This is confirmed by the existing works that exact keyword match is still the most important signal of relevance in ad-hoc retrieval (Fang and Zhai, 2006; Wu et al., 2007; Eickhoff et al., 2015; Guo et al., 2016a,b).

For document filtering, the seed words for a category are expected to convey the conceptual meaning of the latter. It is impossible to list all the words to fully cover the relevant documents of a category. Therefore, it is essential to capture the conceptual relevance for zero-shot document filtering. The classical retrieval models simply estimate the relevance based on the query keyword matching, which is far from capturing the conceptual relevance. The existing deep relevance models for ad-hoc retrieval utilize the statistics of the hard/soft-match signals in terms of cosine similarity between two word embeddings (Guo et al., 2016a; Xiong et al., 2017). However, the scalar information like cosine similarity between two embedding vectors is too coarse or limited to reflect the conceptual relevance. On the contrary, we believe that the embedding features could provide rich knowledge towards the conceptual relevance. A key challenge is to endow DAZER a strong generalization ability to also successfully extract the relevance signals for unseen categories. To achieve this purpose, we extract the relevance signals based on the hidden feature interactions between the category and each word in the embedding space. Specifically, two element-wise operations are utilized in DAZER: element-wise subtraction and element-wise product. Since these two kinds of interactions represent the relative information encoded in hidden embedding space, we expect that the relevance signal extraction process could generalize well to unseen categories. Firstly, DAZER utilizes a gated convolutional operation with k-max pooling to extract the relevance signals. Then, DAZER abstracts higher-level relevance features through a multi-layer perceptron, which can be considered as a relevance aggregation procedure. At last, DAZER calculates an overall score indicating the relevance between a document and the category. Without further constraints, it is possible for DAZER to encode the bias towards the category-specific features seen during the training (*i.e.,* model overfitting). Therefore, we further introduce an adversarial learning over the output of the relevance aggregation procedure. The purpose is to ensure that the higher-level relevance features contain no category-dependent information, leading to a better zero-shot filtering performance.

To the best of our knowledge, DAZER is the first deep model to conduct zero-shot document filtering. We conduct extensive experiments on two real-world document collections from two different domains (*i.e.,* 20-Newsgroup for topic categorization, and Movie Review for sentiment analysis). Our experimetnal results suggest that DAZER achieves promising filtering performance and performs significantly better than the existing alternative solutions, including state-of-the-art deep relevance ranking methods.

2 Deep Zero-Shot Document Filtering

Figure 1 illustrates the network structure of the proposed DAZER model. It consists of two main components: *relevance signal extraction* and *relevance aggregation*. In the following, we present each component in detail.

2.1 Relevance Signal Extraction

Given a document $d = (w_1, w_2, ..., w_{|d|})$ and a set of seed words $\mathbb{S}_c = \{s_{c,i}\}$ for category c, we first map each word w into its dense word embedding representation $\mathbf{e}_w \in \mathbb{R}^{l_e}$ where l_e denotes the dimension number. The embedding representation is pre-trained by using a representation learning method from an external large text corpus. Since our aim is to capture the conceptual relevance, we simply take the averaged embedding of the seed words to represent a category in the embedding space: $\mathbf{c}_c = 1/|\mathbb{S}_c| \sum_{s \in \mathbb{S}_c} \mathbf{e}_s$.

Interaction-based Representation. It is widely recognized that word embeddings are useful because both syntactic and semantic information of words are well encoded (Mikolov et al., 2013; Pennington et al., 2014). The element-wise hidden feature difference is a kind of relative infor-

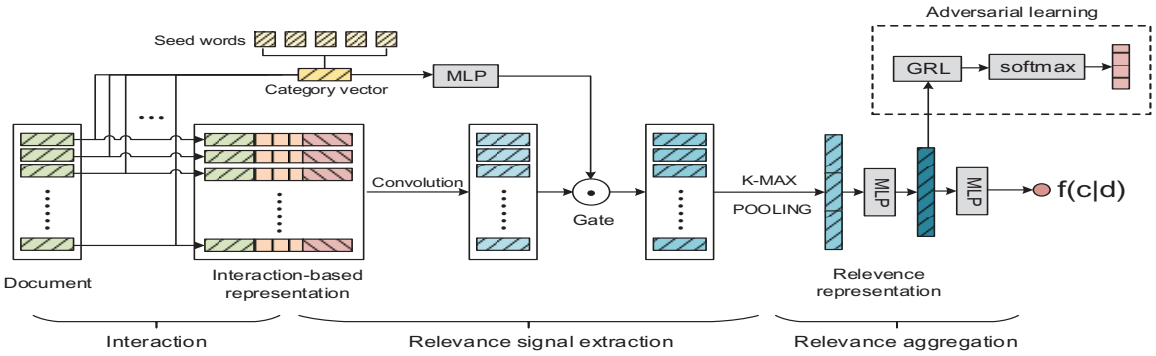

Figure 1: The architecture of DAZER

Examples
$\mathbf{c}_{atheism} - \mathbf{e}_{atheist} \approx \mathbf{c}_{baseball} - \mathbf{e}_{hitter}$
$\mathbf{c}_{autos} - \mathbf{e}_{toyata} \approx \mathbf{c}_{motorcycles} - \mathbf{e}_{yamaha}$
$\mathbf{c}_{baseball} - \mathbf{e}_{stadium} \approx \mathbf{c}_{med} - \mathbf{e}_{hosptial}$
$\mathbf{c}_{religion.misc} - \mathbf{e}_{faith} \approx \mathbf{c}_{med} - \mathbf{e}_{patient}$

Table 1: Examples by using embedding offset.

Examples
$\text{sign}(\mathbf{c}_{mideast} \odot \mathbf{e}_{muslim}) \approx \text{sign}(\mathbf{c}_{med} \odot \mathbf{e}_{doctor})$
$\text{sign}(\mathbf{c}_{space} \odot \mathbf{e}_{orbit}) \approx \text{sign}(\mathbf{c}_{hockey} \odot \mathbf{e}_{espn})$
$\text{sign}(\mathbf{c}_{electronics} \odot \mathbf{e}_{circuit}) \approx \text{sign}(\mathbf{c}_{pc} \odot \mathbf{e}_{controller})$
$\text{sign}(\mathbf{c}_{crypt} \odot \mathbf{e}_{algorithm}) \approx \text{sign}(\mathbf{c}_{space} \odot \mathbf{e}_{burning})$

Table 2: Examples by using element-wise product.

mation that captures the offset bettwen a word and a category in the embedding space. These embedding offsets contain more intricate relationships for a word pair. A well known example is: $\mathbf{e}_{king} - \mathbf{e}_{queen} \approx \mathbf{e}_{man} - \mathbf{e}_{woman}$ (Mikolov et al., 2013). Similar observations are made when we calculate the embedding offset between words and categories. Table 1 lists several interesting patterns observed for the embedding offsets between a category and a word in 20-Newsgroup dataset (ref. Section 3.2 for more details). We can see that the embedding offsets are somehow consistent with a particular relation between the two category-word pairs.

An effective way to measure the relatedness for two words is the inner product or cosine similarity between two corresponding word embeddings. This can be considered as a particular linear combination of corresponding feature products for the two embeddings: $rel(\mathbf{e}_1, \mathbf{e}_2) = \sum_i g(\mathbf{e}_1, \mathbf{e}_2, i)\mathbf{e}_{1,i} \cdot \mathbf{e}_{2,i} = \mathbf{g}(\mathbf{e}_1, \mathbf{e}_2)^T(\mathbf{e}_1 \odot \mathbf{e}_2)$ where $g(\mathbf{e}_1, \mathbf{e}_2, i)$ refers to the weight calculated for i-th dimension, and $\mathbf{g}(\mathbf{e}_1, \mathbf{e}_2) = [g(\mathbf{e}_1, \mathbf{e}_2, 1); ...; g(\mathbf{e}_1, \mathbf{e}_2, l_e)]$, $\odot$ is the element-wise product operation. The element-wise product between two embeddings is also a kind of relative information. The sign of a product of two embeddings in a specific dimension indicates whether the two embeddings share the same polarity in this dimension. And the resultant value manifests to what extent that this agreement/disagreement reaches. It is intuitive that the element-wise product offers some kinds of semantic relations. We conduct the element-wise product for each category-word pair in 20-Newsgroup dataset. Table 2 lists some interesting patterns we observe. The sign(x) function returns 1 when $x \geq 0$, otherwise return -1. Shown in the table, the sign pattern of the element-wise product encodes the relevance information between a category and its related words.

Inspired by these observations, we use these two kinds of element-wise interactions to complement the representation of a word in a document. Specifically, for each word w in document d, we derive its interaction-based representation $\mathbf{e}_w^c$ towards category c as follows:

$$\mathbf{e}_{c,w}^{diff} = \mathbf{c}_c - \mathbf{e}_w \quad (1)$$

$$\mathbf{e}_{c,w}^{prod} = \mathbf{c}_c \odot \mathbf{e}_w \quad (2)$$

$$\mathbf{e}_w^c = [\mathbf{e}_w \oplus \mathbf{e}_{c,w}^{diff} \oplus \mathbf{e}_{c,w}^{prod}] \quad (3)$$

where $\oplus$ is the vector concatenation operation. Note that these two kinds of feature interactions are mainly overlooked by the existing literature. The embedding offsets are used in deriving word semantic hierarchies in (Fu et al., 2014). However, there is no existing work incorporating these two kinds of feature interactions for relevance estimation. Here, we expect that these two kinds of feature interactions can magnify the relevance information regarding the category.

Convolution with k-max Pooling. We utilize

m convolution filters to extract the relevance signals for each word based on its local window of size l in the document. Specifically, after calculating the interaction-based representation $d = (\mathbf{e}^c_1, \mathbf{e}^c_2, ..., \mathbf{e}^c_{|d|})$ for document d and category c, we apply the convolution operation as follows:

$$\mathbf{r}_i = \mathbf{W}_1 \mathbf{e}^c_{i-l:i+l} + \mathbf{b}_1 \tag{4}$$

where $\mathbf{r}_i \in \mathbb{R}^m$ is the hidden features regarding the relevance signal extracted for i-th word, $\mathbf{W}_1 \in \mathbb{R}^{m \times 3l_e(2l+1)}$ and $\mathbf{b}_1 \in \mathbb{R}^m$ are the weight matrix and the corresponding bias vector respectively, $\mathbf{e}^c_{i-l:i+l}$ refers to the concatenation from $\mathbf{e}^c_{i-l}$ to $\mathbf{e}^c_{i+l}$. Both l zero vectors are padded to the begining and the end of the document. With a local window of size l, the convolution operation can extract more accurate relevance information by taking the consecutive words (*e.g.,* phrases) into account. We then apply k-max pooling strategy to obtain the k most active features for each filter. Let $\mathbf{r}^j_{k-max}$ denote the k largest values for filter j, we form the overall relevance signals $\mathbf{r}_d$ extracted by all m filters through the concatenation: $\mathbf{r}_{c,d} = [\mathbf{r}^1_{k-max} \oplus \mathbf{r}^2_{k-max} \cdots \oplus \mathbf{r}^m_{k-max}]$.

Category-specific Gating Mechanism. Given a specific word w, the interaction-based representation $\mathbf{e}^c_w$ for each category c could be very different. Therefore, for a specific local context, the extracted relevance signal from a particular convolution filter could be also distinct for different categories. It is then reasonable to assume that the relevance signals for a specific category are captured by a subset of filters. We propose to identify which filters are relevant to a category through a category-specific gating mechanism. Given category c, category-specific gates $\mathbf{a}_c \in \mathbb{R}^m$ are calculated as follows:

$$\mathbf{a}_c = \sigma(\mathbf{W}_2 \mathbf{e}_c + \mathbf{b}_2) \tag{5}$$

where $\mathbf{W}_2 \in \mathbb{R}^{m \times 3l_e}$ and $\mathbf{b}_2 \in \mathbb{R}^m$ are the weight matrix and bias vectors respectively, $\sigma(\cdot)$ is the *sigmoid* function. With category-specific gating mechanism, Equation 4 can be rewritten as follows: $\mathbf{r}_i = \mathbf{a}_c \odot (\mathbf{W}_1 \mathbf{e}^c_{i-l:i+l} + \mathbf{b}_1)$

Here, $\mathbf{a}_c$ works as *on-off* switches for m filters. While $\mathbf{a}_{c,j} \to 1$ indicates that j-th filter should be turned on to capture the relevance singals under category c to its fullness, $\mathbf{a}_{c,j} \to 0$ indicates that the filter is turned off due to its irrelevance.

This collaboration of the convolution operation and gating mechanism is similar to the Gated Linear Units (GLU) recently proposed in (Dauphin et al., 2017). Given an input $\mathbf{X}$, GLU calculates the output as follow: $\mathbf{h}(\mathbf{X}) = (\mathbf{XW} + \mathbf{b}) \odot \sigma(\mathbf{XV} + \mathbf{c})$ where the first term in the right side refers to the convolution operation and the second term in the right side refers to the gating mechanism. In GLU, both the convolution operation and the gates share the same input $\mathbf{X}$. In contrast, in this work, we aim to identify which filters capture the relevance signals in a category-dependent manner. The experimental results validate that this category-dependent setting brings significant benefit for zero-shot filtering performance (ref. Section 3).

2.2 Relevance Aggregation

The raw relevance signals $\mathbf{r}_{c,d}$ are somehow category-dependent, since the relevant filters are category-dependent. The hidden features regarding the relevance are distilled through a fully-connected hidden layer with nonlinearity:

$$\mathbf{h}_{c,d} = g_a(\mathbf{W}_3 \mathbf{r}_{c,d} + \mathbf{b}_3) \tag{6}$$

where $\mathbf{W}_3 \in \mathbb{R}^{l_a \times 3km}$ and $\mathbf{b}_3 \in \mathbb{R}^{l_a}$ are the weight matrix and bias vector respectively, $g_a(\cdot)$ is the *tanh* function. This procedure can be considered as a relevance aggregation process. Then, the overall relevance score is then estimated as follow:

$$f(c|d) = tanh(\mathbf{w}^T \mathbf{h}_{c,d} + b) \tag{7}$$

where $\mathbf{w} \in \mathbb{R}^{l_a}$ and b are the parameters and bias respective.

2.3 Model Training

Adversarial Learning The hidden features $\mathbf{h}_{c,d}$ are expected to be category-independent. However, there is no guarantee that the category-specific information is not mixed with the relevance information extracted in $\mathbf{h}_{c,d}$. Here, we introduce an adversarial learning mechanism to ensure that no category-specific information can be memorized during the training. Otherwise, the proposed DAZER may not generalize well to unseen categories. Specifically, we introduce an category classifier over $\mathbf{h}_{c,d}$ to calculate the probability that $\mathbf{h}_{c,d}$ belongs to each category seen during the training: $p_{cat}(\cdot|\mathbf{h}_{c,d}) = softmax(\mathbf{W}_4 \mathbf{h}_{c,d} +$

$\mathbf{b}_4$) where $\mathbf{W}_4 \in \mathbb{R}^{C \times l_a}$ and $\mathbf{b}_4 \in \mathbb{R}^C$ are the weight matrix and bias vector for the classifier, C is the number of categories covered by the training set. We aim to optimize parameters $\phi = \{\mathbf{W}_4, \mathbf{b}_4\}$ to successfully classify $\mathbf{h}_{c,d}$ to its true category. Let θ denote the parameters regarding the calculation of $\mathbf{h}_{c,d}$, *i.e.,* $\theta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$, ϕ is optimized to minimize the negative log-likelihood:

$$L_{cat}(\theta, \phi) = \frac{1}{|\mathbb{T}|} \sum_{(d,y)\in\mathbb{T}} -p_{cat}(y|\mathbf{h}_{y,d}) \quad (8)$$

where $\mathbb{T}$ denotes the training set $\{(d, y)\}$ such that document d is relevant to category y. On the other hand, we expect that $\mathbf{h}_{c,d}$ carries no category specific information, such that the classifier can not perform the category classification precisely. Hence, we add the Gradient Reversal Layer (GRL) (Ganin and Lempitsky, 2015; Ganin et al., 2016) between $\mathbf{h}_{c,d}$ and the category classifier. We can consider GRL as a pseudo-function $R_\lambda(\mathbf{x})$:

$$R_\lambda(\mathbf{x}) = \mathbf{x}; \quad \frac{\partial R_\lambda}{\partial \mathbf{x}} = -\lambda \mathbf{I} \quad (9)$$

It means that θ is optimized to make $\mathbf{h}_{c,d}$ indistinguishable by the classifier. In Equation 9, parameter λ controls the importance of the adversarial learning. DAZER is devised to return a relevance score, we utilize the pairwise margin loss for model training:

$$L_{hinge}(\theta, \delta) = \frac{1}{|\mathbb{T}|} \sum_{(d,y)\in\mathbb{T}} \max(0, \Delta - f(y|d) + f(y|d_y^-)) \quad (10)$$

where document d_y^- is the negative sample for category y, Δ is the margin and set to be 1 in this work, and $\delta = \{\mathbf{w}, b\}$. Overall, the proposed DAZER is an end-to-end neural network model. The parameters $\Theta = \{\theta, \phi, \delta\}$ are optimized via back propagation and stochastic gradient descent. Specifically, we utilize Adam (Kingma and Ba, 2014) algorithm for parameter update over mini-batches. The final objective loss used in the training is as follow:

$$L(\Theta) = L_{hinge}(\theta, \delta) + L_{cat}(\theta, \phi) + \lambda_\Theta \|\Theta\|_2 \quad (11)$$

where λ_Θ controls the importance of the regularizaton term.

Label	Seed Words
very negative	bad, horrible, negative, disgusting
negative	bad, confused, unexpected, useless, negative
neutral	normal, moderate, neutral, objective, impersonal
positive	good, positive, outstanding, satisfied, pleased
very positive	positive, impressive, unbelievable, awesome

Table 3: Seed words selected for Movie Review.

3 Experiment

In this section, we conduct experiments on two real-world document collections to evaluate the effectiveness of the proposed DAZER[1].

3.1 Existing Alternative Methods

Here, we compare the proposed DAZER against the following alternative solutions.

BM25 Model: BM25 is a widely known retrieval model based on keyword matching (Robertson and Walker, 1994). The default parameter setting is used in the experiments.

DSSM: DSSM utilizes a multi-layer perceptron to extract hidden representations for both the document and the query (Huang et al., 2013). Then, cosine similarity is calculated as the relevance score based on the representation vectors. Since we use pre-trained word embeddings from a large text corpus, we choose to replace the letter-tri-grams representation with the word embedding representation instead. We use the recommended network setting by its authors.

DRMM: DRMM calculates the relevance based on the histogram information of the semantic relatedness scores between each word in the document and each query word (Guo et al., 2016a). The recommended network setting (*i.e.,* LCH×IDF) and parameter setting are used.

K-NRM: K-NRM is a kernel based neural model for relevance ranking based on word-level hard/soft matching signals (Xiong et al., 2017). We use the recommended setting as in their paper.

DeepRank: DeepRank is a neural relevance ranking model based on the query-centric context (Pang et al., 2017). The recommended setting is used for evaluation.

Seed-based Support Vector Machines (SSVM): We build a seed-driven training set by labeling a training document with a category if the document

[1]The implementation is available at https://github.com/WHUIR/DAZER

contains any seed word of that category. Then, we adopt a one-class SVM implemented by sklearn[2] for document filtering[3]. The optimal performance is reported by tuning the hyper-parameter.

3.2 Datasets and Experimental Setup

20-Newsgroup (20NG)[4] is a widely used benchmark for document classification research (Li et al., 2016). It consists of approximately 20K newsgroup articles from 20 different categories. The *bydate* version with $18,846$ documents is used here. As provided, the training set and test set contain 60% and 40% documents respectively.

Movie Review[5] is a collection of movie reviews in English (Pang and Lee, 2005). The scale dataset v1.0 is used in the experiments. Based on the numerical ratings, we split these reviews into five sentiment labels: *very negative, negative, neutral, positive* and *very positive*, which contains 167, 1030, 1786, 1682, 341 reviews respectively. For each sentiment label, we randomly split the reviews into a training set (80%) and a test set (20%).

Since our work targets at zero-shot document filtering for unseen categories, the word embeddings pre-trained by Glove over a large text corpus with total 840 billion tokens[6] are used across all the methods and the two datasets. The dimension of the word embeddings is $l_e = 300$. No further word embedding fine-tuning is applied. For both datasets, the stop words are removed firstly. Then, all the words are converted into their lowercased forms. We further remove the words whose word embeddings are not supported by Glove.

Evaluation Protocol. With the specified unseen categories, we take all the training documents of the other categories to train a model. Then, all documents in the test set are used for evaluation. For each unseen category, the task is to rank the documents of that category higher than the others. Here, we choose to report mean average precision (MAP) for performance evaluation. MAP is a widely used metric to evaluate the ranking quality. The higher the relevant documents are ranked, the larger the MAP value is, which means a better filtering performance. For all neural networks based models, the training documents from one randomly sampled training category work as the validation set for early stop. We report the averaged results over 5 runs for all the methods (excluding SSVM and BM25). The statistical significance is conducted by applying the student t-test.

Seed Word Selection. For 20NG dataset, we directly use the seed words[7] manually compiled in (Song and Roth, 2014). These seed words are selected from the category descriptions and widely used in the works of dataless text classification (Song and Roth, 2014; Chen et al., 2015; Li et al., 2016). For Movie Review, following the seed word selection process (*i.e.,* assisted by standard LDA) proposed in (Chen et al., 2015), we manually select the seed words for each sentiment label. Table 3 lists the seed words selected for each sentiment label for Movie Review dataset. There are on average 5.2 and 4.6 seed words for each category over 20NG and Movie Review respectively. It is worthwhile to highlight that no category information is exploited within the seed word selection process.

Parameter Setting. For DAZER, the number of convolution filters is $m = 50$ and $k = 3$ is used for k-max pooling. The dimension size for relevance aggregation is $l_a = 75$. The local window size l is set to be 2. The learning rate is 0.00001. The models are trained with a batch size of 16 and $\lambda_\Theta = 0.0001$, $\lambda = 0.1$.

3.3 Performance Comparison

For 20NG dataset, we randomly create 9 document filtering tasks which cover 10 out of 20 categories. For Movie Review, we take each sentiment label as an unseen category for evaluation. Table 4 lists the performance of 7 methods in terms of MAP for these filtering tasks. Here, we make the following observations.

First, the proposed DAZER significantly achieves much better filtering performance on all 14 tasks across the two datasets. The averaged MAP of DAZER over these 14 filtering tasks is 0.671. Note that only 5.2 and 4.6 seed words are used on average for each task. The second best performer is K-NRM, which achieves the second

[2] http://scikit-learn.org

[3] Signed distance to the separating hyperplan is used for ranking documents.

[4] http://qwone.com/~jason/20Newsgroups/

[5] The Movie Review dataset is available at http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6] https://nlp.stanford.edu/projects/glove/

[7] The seed words are available at https://github.com/WHUIR/STM

Dataset	Category	DAZER	DRMM	K-NRM	DeepRank	DSSM	SSVM	BM25
20NG	pc	**0.535**	0.382†	0.369†	0.144†	0.222†	0.117	0.313
	med	**0.826**	0.662†	0.645†	0.033†	0.192†	0.104	0.403
	baseball	**0.764**	0.731†	0.735†	0.294†	0.373†	0.291	0.414
	space	**0.780**	0.593†	0.671†	0.285†	0.142†	0.140	0.641
	med-space	**0.805**	0.640†	0.666†	0.101†	0.174†	0.122	0.522
	atheism-electronics	**0.464**	0.242†	0.346†	0.418†	0.219†	0.132	0.263
	christian-mideast	**0.712**	0.662†	0.657†	0.298†	0.327†	0.161	0.579
	baseball-hockey	**0.782**	0.642†	0.736†	0.332†	0.135†	0.438	0.444
	pc-windowx-electronics	**0.489**	0.274†	0.379†	0.183†	0.278†	0.120	0.314
Movie Review	very negative	**0.290**	0.119†	0.114†	0.097†	0.216†	0.080	0.134
	negative	**0.807**	0.528†	0.557†	0.423†	0.478†	0.236	0.090
	neutral	**0.798**	0.764†	0.749†	0.686†	0.678†	0.365	0.007
	positive	**0.862**	0.696†	0.706†	0.655†	0.753†	0.300	0.090
	very positive	**0.479**	0.250†	0.339†	0.217†	0.271†	0.063	0.066
Avg		**0.671**	0.513	0.548	0.298	0.318	0.191	0.306

Table 4: Performance of the 7 methods for zero-shot document filtering in terms of MAP. The best and second best results are highlighted in boldface and underlined respectively, on each task. † indicates that the difference to the best result is statistically significant at 0.05 level. Avg: averaged MAP over all tasks.

best on 7 tasks. Overall, the averaged performance gain for DAZER over K-NRM is about 30.8%.

Second, We observe that DSSM performs signficantly better for sentiment analysis than for topic categorization. As discussed in Section 4, DSSM is designed to perform semantic matching. Compared with topic categorization, sentiment analysis is more like a semantic matching task. SSVM delivers the worst performance on both datasets. This illustrates that the quality of the labeled documents is essential for supervised learning techniques. Apparently, recruiting training documents with the provided seed words in a simple fashion is error-prone. We also note that BM25 achieves inconsistent performance over the two kinds of tasks. It performs especially worse for sentiment analysis. This is reasonable because there are more diverse ways to express a specific sentiment. It is hard to cover a reasonable proportion of documents with limited number of sentimental seed words. In comparison, the proposed DAZER obtains a consistent performance for both topic categorization and sentiment analysis.

3.4 Analysis of DAZER

Component Setting. Here, we further discuss the impact of different component settings of DAZER on both 20NG and Movie Review datasets. Table 5 and 6 report the impacts of each component setting via an ablation test on the two datasets respectively. We can see that each component brings significantly positive benefit for document filtering. First, we can see that either element-wise subtraction or product contributes signifcantly to the performance improvement. Specifically, from Table 6, we can see that both the element-wise subtraction and element-wise product play equally on Movie Review dataset. On the other hand, it is observed that DAZER experiences significantly a much larger performance degradation on 20NG dataset. For example, a MAP of only 0.154 is achieved when $\mathbf{e}_{c,w}^{prod}$ is excluded from DAZER for the filtering task *space*. A much severer case is for the filtering task *baseball-hockey*. By excluding $\mathbf{e}_{c,w}^{prod}$, the MAP performance of DAZER is reduced from 0.782 to 0.045. That is, the element-wise product is more critical for extracting relevance signals for topical categorization. We also observe that these two hidden feature interactions together play a more important role for DAZER. For example, without both $\mathbf{e}_{c,w}^{diff}$ and $\mathbf{e}_{c,w}^{prod}$, DAZER only achieves a MAP of 0.126 for filtering task *space*. The large performance deterioration is also observed for other filtering tasks on 20NG dataset.

Either adversarial learning or category-specific gate mechanism enhances the filtering performance of DAZER, which validates the effectiveness of the two components for enhancing con-

Setting	pc	med	baseball	space	med-space	atheism-electronics	christian-mideast	baseball-hockey	pc-windowx-electronics
DAZER	**0.535**	**0.826**	**0.764**	**0.780**	**0.805**	**0.464**	**0.712**	0.782	**0.489**
- $\mathbf{e}_{c,w}^{diff}$	0.524	0.810	0.755	0.785	0.802	0.454	0.705	**0.788**	0.462
- $\mathbf{e}_{c,w}^{prod}$	0.219	0.043	0.200	0.154	0.139	0.217	0.244	0.045	0.141
- Gate	0.518	0.819	0.715	**0.780**	0.803	0.443	0.695	0.784	**0.489**
- Adv	0.531	0.819	0.749	0.775	0.795	0.458	0.701	0.779	0.485

Table 5: Impact of different settings for DAZER on 20NG. The best results are highlighted in boldface. - $\mathbf{e}_{c,w}^{diff}$: no element-wise subtraction; - $\mathbf{e}_{c,w}^{prod}$: no element-wise product; - Gate: no category-specific gate mechanism; - Adv: no adversarial learning.

Setting	very negative	negative	neutral	positive	very positive
DAZER	**0.290**	**0.807**	**0.798**	**0.862**	**0.479**
- $\mathbf{e}_{c,w}^{diff}$	0.246	0.773	0.776	0.847	0.453
- $\mathbf{e}_{c,w}^{prod}$	0.258	0.779	0.785	0.847	0.430
- Gate	0.278	0.755	0.785	0.848	0.429
- Adv	0.261	0.779	0.776	0.827	0.444

Table 6: Impact of different settings for DAZER on Movie Review. The best results are highlighted in boldface. - $\mathbf{e}_{c,w}^{diff}$: no element-wise subtraction; - $\mathbf{e}_{c,w}^{prod}$: no element-wise product; - Gate: no category-specific gate mechanism; - Adv: no adversarial learning.

ceptual relevance extraction. Also, without using adversarial learning, DAZER still achieves much better filtering performance than the existing baseline methods compared in Section 3.3. This observation is also held on 20NG dataset. This further validates that the two kinds of hidden feature interactions indeed encode rich knowledge towards the conceptual relevance.

Impact of Seed Words. It has been recognized that the less seed words incur worse document classification performance in the existing dataless document classification techniques (Song and Roth, 2014; Chen et al., 2015; Li et al., 2016). Following these works, we also use the words appearing in the category name of 20NG dataset as the corresponding seed words[8]. There are on average 2.75 seed words for a category of 20NG. Table 7 reports the MAP performace of each method on 20NG dataset. The experimental results show that all methods investigated in Section 3.3 experience signficant performance degradation for most filtering tasks. We plan to incorporate the pseudo-relevance feedback into DAZER to tackle the scarcity of the seed words. One possible solution is to enrich the architecture of DAZER to allow few-shot document filtering. That is, the filtering decisions of high-confidence are utilized to derive more seed words for better filtering performance.

[8]The seed words based on the category name are available at https://github.com/WHUIR/STM

4 Related Work

Document filtering is the task to separate relevant documents from the irrelevant ones for a specific topic (Robertson and Soboroff, 2002; Nanas et al., 2010; Gao et al., 2013, 2015; Proskurnia et al., 2017). Both ranking and classification based solutions have been developed (Harman, 1994; Robertson and Soboroff, 2002; Soboroff and Robertson, 2003). In earlier days, a filtering system is mainly devised to facilitate the document retrieval for the long-term information needs (Mostafa et al., 1997). The term-based pattern mining techniques are widely developed to perform document filtering. A network-based topic profile is built to exploit the term correlation patterns for document filtering (Nanas et al., 2010). Frequent term patterns in terms of fine-grained hidden topics are proposed in (Gao et al., 2013, 2015) for doucment filtering. Very recently, frequent term patterns are also utilized to perform event-based microblog filtering (Proskurnia et al., 2017). However, these approaches are all based on supervised-learning, which requires a significant amount of positive documents for each topic. In the era of big data, the information space and new information needs are continuously growing. Retrieval of the relevance information in a short response time becomes a fundamental need. Recently, many works have been proposed to conduct document filtering in an entity-centric manner (Frank et al., 2012; Balog and Ramampiaro, 2013; Zhou and Chang, 2013; Reinanda et al., 2016). The task is to identify the documents relevant to a specific entity that is well defined in an

Dataset	Category	DEZA	DRMM	KNRM	DeepRank	DSSM	SSVM	BM25
20NG	pc	**0.316**	0.170	0.144	0.104	**0.316**	0.057	0.092
	med	**0.831**	0.369	0.267	0.183	0.089	0.040	0.000
	baseball	**0.519**	0.315	0.301	0.299	0.419	0.066	0.161
	space	**0.641**	0.337	0.326	0.414	0.212	0.049	0.329
	med-space	**0.670**	0.348	0.331	0.279	0.076	0.044	0.165
	atheism-electronics	0.359	0.266	0.253	**0.499**	0.141	0.042	0.091
	christian-mideast	0.564	**0.582**	0.492	0.196	0.418	0.061	0.093
	baseball-hockey	**0.577**	0.409	0.391	0.336	0.154	0.061	0.194
	pc-windowx-electronics	**0.346**	0.176	0.194	0.185	0.227	0.067	0.124

Table 7: Performance of the 7 methods for zero-shot document filtering in terms of MAP. The words appearing in the category name are used as the seed words. The best and second best results are highlighted in boldface and underlined respectively, on each task.

external knowledge base. Specifically, Balog and Ramampiaro (2013) examine the choice of classification against ranking approaches. They found that ranking approach is more suitable for the filtering task. Following this conclusion, we formulate the zero-shot document filtering as a relevance ranking task. Many information needs may not be well represented by a specific entity. Hence, these entity-centric solutions are restricted to knowledge base related tasks.

Many ad-hoc retrieval models can be used to perform zero-shot document filtering. Indeed, traditional term-based document filtering approaches utilize many term-weighting schemes developed for ad-hoc retrieval. Traditional ad-hoc retrieval models mainly estimate the relevance based on keyword matching. BM25 (Robertson and Walker, 1994) can be considered as the optimal practice in this line of literature. The recent advances in word embedding offer effective learning of word semantic relations from a large external corpus. Several neural relevance ranking models are proposed to preform ad-hoc retrieval based on word embeddings. Both K-NRM (Xiong et al., 2017) and DRMM (Guo et al., 2016a) estimate the relevance based on the macro-statistics of the hard/soft-match signals in terms of cosine similarity between two word embeddings. DeepRank (Pang et al., 2017) first measures the relevance signals from the query-centric context of each query keyword matching point through convolutional operations. Then, RNN based networks are adopted to aggregate these relevance signals. These works achieve significantly better retrieval performance than the keyword matching based solutions and represent the new state-of-the-art. The relevance between a query and a document can also be considered as a matching task between two pieces of text. There are many deep matching models, *e.g.,* DSSM (Huang et al., 2013), ARC-II (Hu et al., 2014), MatchPyramid (Pang et al., 2016), Match-SRNN (Wan et al., 2016). These models are mainly developed for some specific *semantic matching* tasks, *e.g.,* paraphrase identification. Therefore, information like grammatical structure or sequence of words are often taken into consideration, which is not applicable to seed word based zero-shot document filtering.

5 Conclusion

In this paper, we propose a novel **d**eep relev**a**nce model for **z**ero-shot document filt**er**ing, named DAZER. To enable DAZER to capture conceptual relevance and generalize well to unseen categories, two kinds of feature interactions, a gated convolutional network and an category-independent adversarial learning are devised. The experimental results over two different tasks validate the superiority of the proposed model. In the future, we plan to enrich the architecture of DAZER to allow few-shot document filtering by incorporating several labeled examples.

6 Acknowledgement

This research was supported by National Natural Science Foundation of China (No.61502344), Natural Science Foundation of Hubei Province (No.2017CFB502), Natural Scientific Research Program of Wuhan University (No.2042017kf0225). Chenliang Li is the corresponding author.

References

Krisztian Balog and Heri Ramampiaro. 2013. Cumulative citation recommendation: classification vs. ranking. In *SIGIR*. pages 941–944.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*. pages 830–835.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John A. Carroll. 2015. Dataless text classification with descriptive LDA. In *AAAI*. pages 2224–2231.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *ICML*. pages 933–941.

Gregory Druck, Gideon S. Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR*. pages 595–602.

Carsten Eickhoff, Sebastian Dungs, and Vu Tran. 2015. An eye-tracking study of query reformulation. In *SIGIR*. pages 13–22.

Hui Fang and ChengXiang Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR*. pages 115–122.

John R. Frank, Max Kleiman-Weiner, Daniel A. Roberts, Feng Niu, Ce Zhang, Christopher Ré, and Ian Soboroff. 2012. Building an entity-centric stream filtering test collection for TREC 2012. In *TREC*.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *ACL*. pages 1199–1209.

Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*. pages 1180–1189.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17:59:1–59:35.

Yang Gao, Yue Xu, and Yuefeng Li. 2013. Pattern-based topic models for information filtering. In *ICDM Workshops*. pages 921–928.

Yang Gao, Yue Xu, and Yuefeng Li. 2015. Pattern-based topics for document modelling in information filtering. *IEEE Trans. Knowl. Data Eng.* 27(6):1629–1642.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016a. A deep relevance matching model for ad-hoc retrieval. In *CIKM*. pages 55–64.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016b. Semantic matching by non-linear word transportation for information retrieval. In *CIKM*. pages 701–710.

Donna Harman. 1994. Overview of the third text retrieval conference (TREC-3). In *TREC*. pages 1–20.

Swapnil Hingmire and Sutanu Chakraborti. 2014. Topic labeled text classification: A weakly supervised approach. In *SIGIR*. pages 385–394.

Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and Sutanu Chakraborti. 2013. Document classification by topic labeling. In *SIGIR*. pages 877–880.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*. pages 2042–2050.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. pages 2333–2338.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. 2016. Effective document labeling with very few seed words: A topic model approach. In *CIKM*.

Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2004. Text classification by labeling words. In *AAAI*. pages 425–430.

Tomas Mikolov, Kai Chen, Greg Corrada, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Javed Mostafa, Snehasis Mukhopadhyay, Wai Lam, and Mathew J. Palakal. 1997. A multilevel approach to intelligent information filtering: Model, system, and evaluation. *ACM Trans. Inf. Syst.* 15(4):368–399.

Nikolaos Nanas, Manolis Vavalis, and Anne N. De Roeck. 2010. A network-based model for high-dimensional information filtering. In *SIGIR*. pages 202–209.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*. pages 115–124.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *AAAI*. pages 2793–2799.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *CIKM*. pages 257–266.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. pages 1532–1543.

Julia Proskurnia, Ruslan Mavlyutov, Carlos Castillo, Karl Aberer, and Philippe Cudré-Mauroux. 2017. Efficient document filtering using vector space topic expansion and pattern-mining: The case of event detection in microposts. In *CIKM*. pages 457–466.

Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2016. Document filtering for long-tail entities. In *CIKM*. pages 771–780.

Stephen E. Robertson and Ian Soboroff. 2002. The TREC 2002 filtering track report. In *TREC*.

Stephen E. Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*. pages 232–241.

Ian Soboroff and Stephen E. Robertson. 2003. Building a filtering test collection for TREC 2002. In *SIGIR*. pages 243–250.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *AAAI*. pages 1579–1585.

Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial RNN. In *IJCAI*. pages 2922–2928.

Ho Chung Wu, Robert W. P. Luk, Kam-Fai Wong, and K. L. Kwok. 2007. A retrospective study of a hybrid document-context based retrieval model. *Inf. Process. Manage.* 43(5):1308–1331.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*. pages 55–64.

Mianwei Zhou and Kevin Chen-Chuan Chang. 2013. Entity-centric document filtering: boosting feature mapping through meta-features. In *CIKM*. pages 119–128.