Vocabulary Filtering for Term Weighting in Archived Question Search

Zhao-Yan Ming, Kai Wang, and Tat-Seng Chua

Department of Computer Science,
School of Computing, National University of Singapore
{mingzy,kwang,chuats}@comp.nus.edu.sg

Abstract. This paper proposes the notion of vocabulary filtering in a term weighting framework that consists of three filters at the document level, collection level, and vocabulary level. While term frequency and document frequency along with their variations are respectively the dominant term weighting factors at the document level and collection level, vocabulary level factors are seldom considered in current models. In a way, stopword removal can be seen as a vocabulary level filter, but it is not well integrated into the current term-weighting models. In this paper, we propose a vocabulary filtering and multi-level term weighting model by integrating point-wise divergence based measure into the commonly used TF-IDF model. With our proposed model, the specificity of the vocabulary is captured as a new factor in term weighting, and stopwords are naturally handled within the model rather than being removed according to a separately constructed list. Experiments conducted on searching for similar questions in a large community-based question answering archive show that: (a)our proposed term weighting model with multiple levels is consistently better than those with single level for retrieval task; (b)the proposed vocabulary filter well distinguishes salient and trivial terms, and can be utilized to construct stopword lists.

1 Introduction

As large Community-based Question Answering (cQA) archives are built up through user collaboration, the knowledge is accumulated and made ready for sharing. Research on archived questions has emerged recently [5,14,15]. An application that facilitates knowledge sharing and diversity maintaining is *Archived Question Search* (AQS). It is a function that makes the huge resource reusable by returning relevant answered questions given a new question as a query. If good matches are found, the lag time involved with waiting for a personal response can be avoided, thus improving user satisfaction and avoiding repeating questions.

As a specific application of IR, AQS in cQA repository is distinct from the search of web pages or news articles because it deals with long queries and short documents that are both in the form of questions (for simplicity, we call query questions in AQS as queries, and candidate questions to be searched as documents thereafter):

Long query: each AQS queries consist of natural language sentences that are supposed to be understood and answered by other community members. This kind of queries is

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 383–390, 2010.

usually longer, noisier, and more verbose than keyword queries. Thus the salient terms and trivial terms are weaved together and the information needs are usually more specific.

Short document: AQS documents are essentially the same as its queries since both are cQA questions. Shorter document length means that most terms might appear only once in a document, resulting in term frequency (tf) approximates a weak binary factor.

This characteristic of cQA data makes the existing term weighting schemes, such as TF-IDF model [13], Okapi BM25 [10], and Divergence From Randomness [2], less capable if directly applied. The major difficulty is that documents and collections statistics are not adequate to provide enough information. The proper functioning of the existing term weighting schemes is under the assumption that documents are long and queries are short. The same difficulty may also be encountered by other forms of community content,like blogs and forums, which have the concise and noise-prone nature.

This motivates us to explore beyond the document and collection. We propose the notion of vocabulary filtering as a complementary dimension of term weighting. By definition, a vocabulary refers to the body of words used on a particular setting or in a particular domain. Although different vocabularies may share a similar set of terms, their respective weights are specific. Given the underlying setting or domain, the weights can be determined, independent of any specific collection and document that instantiates the vocabulary. In a way, stopword removal can be seen as a simple binary vocabulary filter, in that it assigns 0 or 1 score to a term in additional to a weighting scheme.

We propose to measure term saliency in a vocabulary by estimating a heuristic evaluation function that accepts terms' point-wise divergence feature as input. The assumption under the point-wise divergence feature is that terms that have distinct distributions in a specific vocabulary vs. the general vocabulary are important. For instance, we expect "ipod" to have a much higher frequency in a vocabulary about $music \ \& \ music\, players$ than it does in the general vocabulary, whereas the universal stopword "the" would have similar frequencies in the two vocabularies. The vocabulary filter, in the form of a heuristic term saliency evaluation function, is integrated into the existing term weighting schemes as an enhancement.

2 Related Work

Archived Question Search (AQS) in cQA repository was investigated recently by [5] and [15] using translation-based language model. The translation probability trained on similar collections can be seen as a form of collection-level filtering of term weights. We attribute the success of their proposed model to the integration of the collection-level evidence into the document-level language model.

In related research on term weighting models, the TF-IDF model has been widely used and accepted. Recently, the justification and interpretation of TF-IDF has been studied in [1,4,12], from the perspectives of information theory, probabilistic language modeling, binary independence retrieval, and Poisson distribution. [11] tried to interpret the Okapi BM25 model from the perspective of poisson model, the language model, the TF-IDF model, and a Divergence From Randomness model. However, none of these efforts attempt to separate the evidences between document-level and the collection-level in term weighting. Since our investigation is focused on the vocabulary-level evidences, it further raises question on the roles of document level, collection level as well as vocabulary level analysis in term weighting and IR effectiveness.

As for the vocabulary-level filtering for term weighting, the building of a customized stopword list [7,8] and the extraction of domain-specific keywords [3] can be seen as the most relevant work. [6] evaluated the interestingness of a term using the KL divergence and the JS divergence from the distribution of the human interested corpora. This work inspires the method we use for vocabulary filtering.

3 Proposed Vocabulary-Level Filter

Our goal to build a vocabulary-level filter is to quantitatively measure term saliency for a specific vocabulary. We thus emphasize the specificity of vocabularies in constructing the vocabulary-level filters. Term distribution in a specific collection is biased as compared to that in a general collection. This specificity of term distribution in a collection reflects the specificity of its vocabulary, and enables us to highlight the specific important terms for a vocabulary.

3.1 Divergence Feature for Vocabulary Filtering

To capture the vocabulary level term importance, we propose to take a novel point-wise divergence feature for each individual term, rather than divergence of two distributions. We see the term distribution of a vocabulary as the background knowledge to instantiate vocabulary filtering. More broadly, the combining of all vocabularies consists of a general background that can be used to compare against a specific vocabulary.

Jensen-Shannon(JS) divergence is a well adopted distance measure between two probability distributions. It is defined as the mean of the relative entropy of each distribution to the mean distribution, with the following formula:

$$D_{JS}(S||G) = \frac{1}{2}\sum p_s \log \frac{p_s}{\frac{1}{2}(p_s + p_g)} + \frac{1}{2}\sum p_g \log \frac{p_g}{\frac{1}{2}(p_s + p_g)} \tag{1}$$

where S and G denote the specific and general vocabularies, and $p_s(t_i)$ and $p_g(t_i)$ denote their corresponding probability distribution.

As we evaluate the divergence at term level rather than at the whole sample set, we examine the point-wise function as follows:

$$d_{JS}(t) = \frac{p_s(t) \log \frac{2p_s(t)}{p_s(t)+p_g(t)} + p_g(t) \log \frac{2p_g(t)}{p_s(t)+p_g(t)}}{2} \tag{2}$$

The point-wise JS function is an appropriate choice since it is symmetric and ranges over $\Re+$. Specifically, $d_{JS}(t)$ assigns a point-wise divergence score to term t highest, when either $p_s(t)$ is much higher than $p_g(t)$, or $p_g(t)$ is much higher than $p_s(t)$, which means the specialized terms in the vocabulary and generally recognized content representative terms are ranked high; lower, when $p_s(t)$ and $p_g(t)$ get closer to each other. These properties suggest that d_{JS} emphasizes divergence at both the most frequent terms in the specific vocabulary and the most frequent terms in the general vocabulary. $p_s(t)$ and $p_g(t)$ are estimated using the Maximum Likelihood Estimator over the specific vocabulary and the general vocabulary respectively.

3.2 Estimating Term Saliency from Divergence Feature

Given a point-wise divergence feature, we aim to estimate the term saliency score, which can be integrated with any existing term weighting scheme. More specifically, we define a mapping function $f_v : d_{JS} \rightarrow W_v$, which produces as output an estimation W_v(denotes the term saliency score on $\Re+$) given d_{JS} as input.

We propose a heuristic evaluation function based on logistic function $L(x)$ as below.

$$f_v(x) = 1 + \tau L(x), \tag{3}$$

where $L(x) = \frac{1}{1+e^{-x}}$ is the logistic function that maps $\Re+$ to $[0.5, 1)$ monotonically.

Logistic function grows more slowly as $|x|$ increases. This property enables us to control the rate of normalization by shifting the curve along the horizontal axis. Thus a tuning parameter α is introduced.Equation 4 represents the final form of the the heuristic evaluation function of vocabulary level term saliency.

$$f_v(d_{JS}) = 1 + \tau \frac{1}{1 + e^{-(d_{JS}+\alpha)}} \tag{4}$$

where the parameters τ and α are tuned in Section 5.2. Since there is no direct observation of term saliency available, we tune the parameters by using the retrieval performance as an indirect guidance.

4 Three-Level Filtering for Term Weighting

After discussing the method for term weighting accounts for the vocabulary-level informativeness of a term, we next investigate on how it can be naturally integrated into an IR framework.

Term weighting lies in the core of current bag-of-word IR algorithms. Terms weights discriminate the importance of terms for content representation as document descriptors. The general form of the bag-of-words retrieval function with regards to term weighting is given as:

$$Score(Q, D) = \sum_{t_i \in (Q \cap D)} W(t_i) \tag{5}$$

where t_i is the i^{th} query term that appears in both the query Q and the document D, and $W(\cdot)$ is the term weighting model. Generally, $W(\cdot)$ is a function that takes in evidences such as term frequency, document length, document frequency, *etc.*

Figure 1 illustrates the pipeline of our proposed three-level filtering framework. Given a term t_i as input, the pipeline of three filters outputs $W_t(t_i)$, a quantity that indicates t_i's importance. Accordingly, we break down the term weighting model into three components. The scoring function in Equation (5) is thus rewritten as

$$Score(Q, D) = \sum_{t_i \in (Q \cap D)} f_v(t_i) \times f_c(t_i) \times f_d(t_i) \tag{6}$$

where $f_v(t_i)$, $f_c(t_i)$, and $f_d(t_i)$ are the *Vocabulary-Level Filter*, *Collection-Level Filter*, and *Document-Level Filter* respectively. The sequence of filters in the pipeline is naturally decided by the scope of the filters and the implementation process.

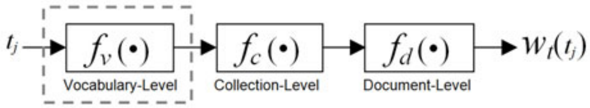

Fig. 1. A general framework for term weighting schemes is represented in a pipeline of three filters, the vocabulary level filter, collection level filter, and document level filter

5 Experiment

5.1 Data Collection and Evaluation Method

We assembled a collection of 325,274 questions posted on Yahoo! Answers(YA) category $Consumer\ Electronics$ from March 2008 to December 2008 using YA APIs[1]. The archived questions have an average length of approximately 60 words, consisting of "subject" and "content". The statistic for replicating the term distribution in the general vocabulary was acquired from a project called Web Term Document Frequency and Rank, a joint effort of the UC Berkeley and Stanford WebBase Projects[2].

For evaluation, 100 questions were assembled by randomly selecting questions from the whole collection. 93 questions were finally used after manually removing the noisy and redundant ones. The top 20 results of the 93 queries by all the experimented models were labeled by two independent assessors to be relevant or not. The evaluation system, as well as the testing set and the archive, are publicly accessible[3].

We use Terrier [9] for indexing and retrieving, and Porter Stemmer to stem the cQA collection, the queries, and the general web vocabulary. The evaluation metrics are Mean Average Precision(MAP) and Mean Reciprocal Rank (MRR).

5.2 Archived Question Search with Vocabulary Filters

Experimental Setup. In this suit of experiment, we use vocabulary filtering to replace the role of stopword lists in order to test the overall quality of the new retrieval function. The efficiency aspect of stopword removal is not considered here. The comparison systems are (1)$D.(tf\ )$; (2) $C.D.(idf\text{-}tf)$; (3) $V.D.\ (js\text{-}tf)$; and (4) $V.C.D.\ (js\text{-}idf\text{-}tf)$, where $V.$ denotes the proposed vocabulary-level filter $f_v(d_{JS})$; $C.$ denotes the collection-level filtering ($f_c(t) = ln(1 + \frac{N}{df})$), $D.$ denotes the document-level filtering ($f_d(t) = 1 + ln\frac{tf}{dl}$).

Parameter Tuning for Term Salience Function. A small set of 20 queries are used for tuning the parameters of the term salience estimation function as in in Equation 4. For simplicity, we fix τ to be 1.0 as it does not influence the shape of the function. Therefore, only the optimal α is studied in this section.

Figure 2 shows the influence of α on MAP for $V.C.D.$ term weighting scheme. At the point $\alpha = 2.0$, the MAP starts to approach its maximum. This is because the logistic

[1] http://developer.yahoo.com/answers/

[2] http://www.comp.nus.edu.sg/∼rpnlpir/

[3] http://www.comp.nus.edu.sg/∼g0601820/aqs/

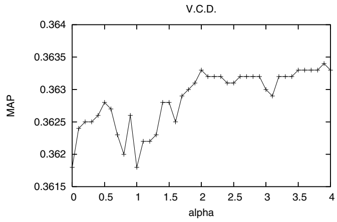

Fig. 2. MAP at different normalization level of α for *V.C.D.*

function begins to saturate around 2 and grows slowly thereafter as the input increases on $\Re+$. This slow growth satisfies the requirement for a deep normalization on $d_{JS}(t)$. It is worth noticing that this optimal α ranges over $[2.0, 2.9]$. This relatively wide range suggests the stability and tolerance of the proposed heuristic term salience estimation function. For the rest of the experiments, α is set to be 2.0.

Table 1. The MAP and MRR of *V.C.D.* and improvement over two baselines *D.* and *C.D.*

MAP		*D.*	*C.D.*
		0.1874	0.3182
V.D.	0.2942	56.99%	-7.54%
V.C.D	0.3622	93.28%	13.83%

MRR		*D.*	*C.D.*
		0.5856	0.6987
V.D.	0.7396	26.30%	5.85%
V.C.D	0.7616	30.05%	9.00%

Overall Results and Discussion. Table 1 presents the overall results in term of MAP and MRR. MAP is based on the top 20 returned results and MRR evaluates the quality of the top results returned. We draw following observations from the table:

1) Vocabulary Filter in conjunction with collection and document level filters (*i.e.* V.C.D), boosts tf and $idf.tf$ significantly in both MAP and MRR. MAP comparison in Table 1 shows that *V.C.D.* improves over *D.* by 93.28%, and *C.D.* by 13.83%. The consistent improvement suggests that vocabulary level evidence is complimentary to collection level and document factors for term weighting. The scale of improvement over *C.* is higher than that over *C.D.*, which indicates *V.D.* is a much stronger baseline than *D.*. In other words, collection level evidence is also critical for measuring term importance. The only negative improvement is *V.D.* over *C.D.*, which shows that *V.D.* is not as effective as the classical $tf.idf$ model. This also suggests that *V.*, *C.*, and *D.* are three orthogonal factors critical for term weighting.

2) Vocabulary Filter improves the top results when the baselines are already very high. By examining MRR comparison in Table 1, we find that two baseline systems both have MRR of over 0.5, which suggests that YA archives have considerable number of similar questions and it is relatively easy to find a similar one with a high ranking. The two systems with *V.*, *i.e.*, *V.D.* and *V.C.D.* both have MRR of over 0.7, which shows that both systems have most of their top retrieval results correct. We also notice that *V.D.* has a higher MRR than *C.D.*, while the latter has higher MAP, which confirms our assertion of the orthogonal of *V.*, *C.*, and *D.*.

Table 2. Highest and lowest ranked 10 terms in $music\ \&\ music\ players$ category by $f_v(d_{JS})$

Top 10				Lowest 10			
1	ipod	6	home	-1	us	-6	at
2	itune	7	servic	-2	of	-7	thi
3	page	8	provid	-3	in	-8	on
4	my	9	mail	-4	by	-9	all
5	song	10	copyright	-5	new	-10	be

5.3 Study on Rank Terms Using Vocabulary Filtering

To examine the effect of the two divergence kernels, we utilize them to rank terms from a subcategory of $ConsumerElectronics$, $i.e.$, the $music\ \&\ music\ players$ question archive. From the top 10 terms in Table 2, we find that the vocabulary filter has successfully captured the salient terms of the recent $music\ \&\ music\ players$ vocabulary, such as "ipod", "itune", and "sync". We may guess that if the archive was collected years earlier, the top terms might be "walkman", "tape" and the like. It suggests that the vocabularies are evolving, or more generally, are specific. We notice that terms like "my" is ranked high, because of the user-collaborative nature of the YA archive. $f_v(d_{js})$ also ranks some *general* terms high, such as "mail" and "copyright". This is because "mail" and "copyright" have high probabilities in the general web vocabulary, but low probabilities in the $music\ \&\ music\ players$ vocabulary. They are considered to be informative though relatively less frequent in the specific vocabulary. This shows that $f_v(d_{JS})$ is capable of capturing term importance in a vocabulary.

The stopwords are expected be among the lowest in a descending ranked list. Left part of Table 2 lists the lowest 10 terms by $f_v(d_{JS})$. Generally this lowest 10 terms meet our expectation of the commonly recognized stopwords. We thus think of eliminating the lowest ranked terms from indexing, as what stopword removal does, for the purpose of improving the efficiency of the whole retrieval system. As a complementary exploration, we construct stopword lists by taking the lowest 5%, 10%, and 15% $f_v(d_{JS})$ ranked terms. In stead of implementing the full-fledged $V.C.D.$ filtering term weighting scheme, we use the 3 stopword lists of different size upon $C.D.$ term weighting scheme and find that the retrieval performance at 5% removal actually improves over those without stopword removal and with standard stopword list removal. Moreover, 10% removal is slightly worse than 5% removal since less terms are used for indexing, but still acceptable considering that the efficiency is improved at a small price.

6 Conclusions

In this paper, we proposed a novel notion of vocabulary-filtering to capture term importance by using the whole vocabulary as the background knowledge. JS divergences are utilized to characterize a specific vocabulary by contrasting its term distribution to that of of a general vocabulary. The normalized vocabulary filters are integrated into a framework that consists of a pipeline of three filters at the document level, collection level, and vocabulary level. Our proposed model has been empirically shown to be significantly better than TF-IDF model in tackling the archived question search problem.

In future work, we plan to explore the use of vocabulary filtering and the three-level term weighting schemes in other text processing tasks like document clustering and categorization.

References

1. Aizawa, A.: An information-theoretic perspective of tf-idf measures. Inf. Process. Manage. 39(1), 45–65 (2003)
2. Amati, G., Joost, C., Rijsbergen, V.: Probabilistic models for information retrieval based on divergence from randomness. ACM TOIS 20, 357–389 (2002)
3. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-manning, C.G.: Domain-specific keyphrase extraction, pp. 668–673. Morgan Kaufmann Publishers, San Francisco (1999)
4. Hiemstra, D.: A probabilistic justification for using tf idf term weighting in information retrieval. International Journal on Digital Libraries 3, 131–139 (2000)
5. Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: Proc. of CIKM, pp. 84–90. ACM, New York (2005)
6. Kor, K.-W., Chua, T.-S.: Interesting nuggets and their impact on definitional question answering. In: Proc. of SIGIR, pp. 335–342. ACM, New York (2007)
7. Lo, R., He, B., Ounis, I.: Automatically building a stopword list for an information retrieval system. In: Proc. of Dutch-Belgian DIR, Utrecht, Netherlands (2005)
8. Makrehchi, M., Kamel, M.S.: Automatic extraction of domain-specific stopwords from labeled documents. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 222–233. Springer, Heidelberg (2008)
9. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A High Performance and Scalable Information Retrieval Platform. In: Proc. of SIGIR Workshop on OSIR (2006)
10. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proc. of SIGIR, pp. 232–241. Springer, New York (1994)
11. Robertson, S.E., Zaragoza, H.: The probabilistic relevance model: Bm25 and beyond. In: Tutorial of SIGIR, ACM, New York (2008)
12. Roelleke, T., Wang, J.: Tf-idf uncovered: a study of theories and probabilities. In: Proc. of SIGIR, pp. 435–442. ACM, New York (2008)
13. Salton, G., Yang, C.: On the specification of term values in automatic indexing. Journal of Documentation 29, 351–372 (1973)
14. Wang, K., Ming, Z., Chua, T.-S.: A syntactic tree matching approach to finding similar questions in community-based qa services. In: Proc. of SIGIR, pp. 187–194. ACM, New York (2009)
15. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proc. of SIGIR, pp. 475–482. ACM, New York (2008)