JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 28, 601-615 (2012)

Short Paper

A Keyword Based Prototype for Web Search Result Diversification*

GU-LI LIN, HONG PENG, QIAN-LI MA, JIA WEI AND JIANG-WEI QIN
School of Computer Science and Engineering
South China University of Technology
Guangzhou, 510006 China

In web search scenario, users often submit short query terms to search engines, expecting to find their desired information in top ranked results. But their queries are so ambiguous that their actual information needs are often unspecified. To satisfy the different information needs, an effective approach is to diversify the top results retrieved for the query. In this paper, we reduce the diversification problem into optimizing the maximum coverage of information facets related to the query, and introduce KED, a novel keyword based prototype for web search result diversification that provides a diverse ranking by selecting documents to cover keywords which belong to different facets underlying the retrieved documents. We evaluated the effectiveness of KED using two public test collections with different kinds of documents. The experiment results show that KED can stably outperform other existing implicit diversification approaches in promoting diversity of top ranked results. Moreover, we show that its effectiveness can be further improved by using high quality keywords.

Keywords: information retrieval, search result diversification, search result re-ranking, document novelty, keyword extraction

1. INTRODUCTION

In the scenario of web search, users usually submit short queries to search engines, expecting to find their desired information in the top retrieved results. But the short queries are often so ambiguous that search engines can't tell what the users' actual information needs are. Even if the interpretations are clear, users may be interested in different facets of information underlying the queries. For example, a query 'Michael Jordan' may refer to different people, such as the famous basketball player, or the famous Machine Learning researcher. And a user searching for the basketball star may wonder his biography, news or other information.

In such situations, most of the current search engines, which rank the retrieved documents by independently considering the relevance of the documents to the query, can not

Received June 26, 2010; revised December 22, 2010; accepted March 23, 2011.
Communicated by Jonathan Lee.
* The work described in this paper was partially supported by Grants from the National Natural Science Foundation of China (Project No. 61070090, 61003174 and 60973083), a grant from NSFC-Guangdong Joint Fund (Project No. U1035004), grants from Natural Science Foundation of Guangdong Province, China (Project No. 9451064101003233, 10252500002000001 and 10451064101004233), and grants from Fundamental Research Funds for the Central Universities (Project No. 2009ZM0125, 2009ZM0189 and 2009ZM0255).

well satisfy the need of diversity. A more sensible way is to provide relevant but diverse top results that cover different facets of information related to the query, so that users can find at least one document related to their needs in the top results.

The general problem of web search result diversification that maximizing the facet coverage or minimizing the facet redundancy by the top results is NP-hard [1]. Most existing works can be categorized as implicit or explicit diversification, according to the way in which they account for the underlying facets covered by the query [2]. Implicit diversification approaches don't depend on the knowledge about the underlying facets. Most of them re-rank the retrieved documents by directly comparing them against one another to reduce redundancy, while others select documents to cover important words for improving diversity. To the contrary, explicit diversification approaches usually directly model the underlying facets, and promote diversity based on estimating the relationship between the documents and the facets.

In this paper, we introduce a novel keyword based prototype for web search result diversification which can be categorized as implicit diversification. But differently from the existing implicit diversification approaches, it doesn't promote diversity by simply comparing the documents against one another, or by selecting documents to cover words with independent importance scores. Instead, it utilizes keywords to connect the retrieved documents and the underlying facets, and provides a diverse top ranking by selecting documents to cover the keywords based on their facet novelty. In particular, it extracts keywords from the retrieved documents, expecting that different underlying facets can be represented by different combination of keywords. And then it explicitly models the relationship of proximity on facet level between the keywords. By doing so, it can model the facet novelty of a single document in face of the keywords already covered by ranked documents during the ranking process.

We evaluated our prototype using two public test collections, and the experiment results show that our proposed prototype can stably outperform other existing implicit diversification approaches. The rest of this paper is organized as follows. Section 2 provides a survey on previous related works. Section 3 details our keyword based prototype. Section 4 presents our experimental settings, while section 5 discusses our main findings. Finally, section 6 presents our conclusions.

2. RELATED WORK

The need of diversity has been realized since early work [3] on information retrieval. In recent years, the diversification problem attracts more and more attentions [4, 5]. The existing approaches can be categorized as either implicit or explicit diversification, according to the way in which they account for the underlying facets of the query [2].

Most implicit diversification approaches directly compare the retrieved documents against one another, under the assumption that similar documents would cover similar facets, and improve diversity by demoting the documents similar with ranked documents. Carbonell *et al*. [6] proposed an influential criterion called maximal marginal relevance (MMR) to reduce redundancy while maintaining query relevance in re-ranked documents. Based on MMR, Zhai *et al*. [7] modeled the query relevance and redundancy in the MMR criterion within the language model framework and proposed several methods based on the

K-L divergence measure and a simple mixture model. Gollapudi *et al*. [8] utilized axiomatic approach to describe the diversification problem. They proposed three optimization objects for diversification, reduced two of them into facility dispersion problems [9] and utilized two well-studied algorithms to solve them. Zhu *et al*. [10] proposed Grasshopper to improve diversity by combining centrality, diversity and user prior in a unified framework of absorbing Markov random walks.

Furthermore, there is another type of implicit diversification approaches which focuses on word space covering. Swaminathan *et al*. [11] proposed Essential Pages method to improve diversity by selecting a subset of retrieved documents that maximizes the information coverage related to a given query. Similar with Essential Pages method, Yue *et al*. [12] proposed a supervised structure learning approach called SVM^{div}, learning the importance of individual words and then selecting the optimal set of documents that covers the largest number of important words.

Differently from the implicit diversification approaches, explicit diversification approaches directly model the underlying facets associated to the query. Agrawal *et al*. [1] proposed a diversification method that tries to maximize the likelihood of finding a relevant document in the top-*k* positions given the categorical information of queries and documents. Santos *et al*. [2, 13] proposed a probabilistic diversification framework named xQuAD, which performs an explicit diversification by exploiting the relationship between the retrieved documents and the uncovered facets.

Our proposed prototype can be categorized as implicit diversification, since it doesn't need to dependent on outside facet resources nor model the underlying facets. But differently from the aforementioned implicit diversification approaches, it doesn't reduce redundancy by directly comparing the documents with each other, or by selecting documents to cover words with independent importance scores. Instead, it firstly extracts keywords from the retrieved documents, and then models the novelty of keywords based on their distance on the facet level. After that, it models the facet novelty of documents based on their coverage of keywords. By doing so, it can select documents covering more different facets to provide a diverse ranking with richer information.

3. DIVERSIFYING WEB SEARCH RESULTS

3.1 The KED Diversification Prototype

In this section, we details our keyword based prototype for web search result diversification.

The *KEyword based Diversification* (KED) prototype follows the general greedy approximation solution for the diversification problem. But differently from the previous works, KED performs an implicit diversification on the retrieved documents for a given query, by utilizing keywords as the basic element of the underlying facets and exploiting the facet novelty of documents based on their coverage of keywords for diverse ranking.

Specifically, since the retrieved documents cover many different facets of information related to the query, we argue that there exists a set of keywords whose different subsets can well represent the different underlying facets. And then we can select a document subset to cover the keywords, so as to cover all the facets. We realize that, between any

two keywords there exists some kind of distance which reflects their relationship of proximity on facet level. We define it as *facet distance*. Intuitively, if keyword A and keyword B belong to the same facet, and keyword C belongs to another facet, then A would have larger facet distance to C than to B. Then, during the ranking process in diversification, selecting documents that cover keywords with small facet distance can't improve diversity. The existed word space covering thods [11, 12] may suffer from this situation, since they select documents according to the importance scores of their words, without considering the relationship between the words in ranked documents and documents to be ranked. Therefore, we proposed KED, which takes into account the facet novelty of the documents to be ranked, based on the facet distance between their keywords and the keywords already covered by ranked documents. During the document selection process, it iteratively selects the document with the largest combination score of facet novelty and query relevance, aiming at providing a diverse ranking with less facet redundancy. The working scheme of KED is described in Algorithm 1.

Algorithm 1 The KED diversification prototype

KED(q, D, KW, k, λ)

1. $SD = \varnothing$
2. $KW_C = \varnothing$
3. select the first document d^* ($d^* \in D$) with certain strategy
4. while $||SD|| < k$ do
5. $SD = SD \cup \{d^*\}$
6. $D = D \backslash \{d^*\}$
7. $KW_C = KW_C \cup KW_{d^*}$
8. if $||KW_C|| == ||KW||$
9. break
10. end if
11. $d^* = \text{argmax}_{d_i}(\lambda R(d_i, q) + (1 - \lambda)NS_d(d_i))$, $\forall d_i \in D$
12. end while
13. return SD

Given a query q, a set D of retrieved documents for q, and a set KW of keywords extracted from D, KED provides a k-document ranked list SD to perform diversification. The key point in KED is the combination score, *i.e.*

$$\lambda R(d_i, q) + (1 - \lambda)NS_d(d_i) \tag{1}$$

where $R(d_i, q)$ is the relevance score of document d_i to query q, and $NS_d(d_i)$ is the facet novelty score of document d_i. λ ($\lambda \in [0, 1]$) is used to give a tradeoff between relevance and novelty.

At the beginning of KED, none of documents is selected ($SD = \varnothing$), and no keywords is covered. KED selects the first document d^* with certain strategy, and then the set KW_{d^*} of keywords that d^* covers is added into the Covered Keyword Set KW_C. After that, the facet novelty scores of the remaining documents are calculated, and the document with the largest combination score in Eq. (1) is ranked at the second position. The procedure is repeated until k documents are selected or all keywords are covered.

3.2 Prototype Implementation

3.2.1 Keyword extraction

Keyword extraction is an important part of our prototype, since we expect that the extracted keyword can be combined to well represent the underlying facets. We utilize suffix array based technique to extract keywords, which has been successfully applied in web search result clustering [14]. The process of our implementation of keyword extraction can be summarized as follows:

- Preprocessing: We preprocess the retrieved documents with Porter stemming [15], stop word marking and text segmentation. In order to extract complete phrases, we do not remove the stop words at this step since they may be parts of phrases. Text segmentation divides text into words and sentences, which is important for phrase extraction, since a phrase extending beyond one sentence is likely to carry little meaning to the user [16].
- Keyword extraction: We utilize suffix array based technique to extract frequent single words and phrases, and then select the keywords from them with rules. Specifically, we use the phrase discovery algorithm proposed in [14], which employs a variant of suffix array. Then, we remove the single words that are stop words and the phrases which begin or end with stop words. Finally, the filtered single words and phrases that exceed the keyword frequency threshold T are chosen as keywords. Compared to a set of single words, phrases have greater descriptive power, as they can retain the relationships of proximity and orders between the words. Additionally, removing the infrequent single words and phrases can reduce the calculating complexity of the method.

3.2.2 Facet novelty of a document

To measure the facet novelty of a document, we define a *document facet novelty score* based on its coverage of keywords in face of the keywords already covered by the ranked documents during the ranking process.

Suppose a set KW of m keywords is extracted from n candidate documents, different underlying facets contain different subsets of the keywords. Intuitively, if two keywords occur in the same documents with closer frequencies more often, they would be more likely to belong to the same facet. Based on this assumption, we define the facet distance between two keywords as:

$$FD(kw_j, kw_w) = \sqrt{\sum_{i=1}^{n} (TF(kw_j, d_i) - TF(kw_w, d_i))^2} \tag{2}$$

where $i \in [1, n]$, $j, w \in [1, m]$, and $TF(kw_j, d_i)$ represents the frequency of keyword kw_j appears in document d_i:

$$TF(kw_j, d_i) = c(kw_j, d_i) / \sum_{w=1}^{m} c(kw_w, d_i) \tag{3}$$

where $c(kw_j, d_i)$ represents the number of times that keyword kw_j appears in document d_i.

The larger facet distance score between two keywords means that they are more novel to each other.

Since documents are composed of keywords, if the most representative keywords of a document are novel, the document is probable to be novel. During the ranking process, the novelty of a keyword is defined as the distance between the keyword and the Covered Keyword Set KW_C, which contains the keywords covered by the ranked documents:

$$NS_w(kw_j) = FD'(kw_j, KW_C) \tag{4}$$

where the distance between keyword kw_j and KW_C is defined as the smallest distance between kw_j and any keyword in KW_C, *i.e.*,

$$FD'(kw_j, KW_C) = \min FD(kw_j, kw_s), \forall kw_s \in KW_C. \tag{5}$$

Moreover, we utilize term frequency to measure the representative power of a keyword for a document, since it is widely accepted that words with frequent occurrence in a document have strong descriptive power for that document. Finally, the facet novelty score of document d_i is given by:

$$NS_d(d_i) = \sum_{j=1}^{m} TF(kw_j, d_i) \times NS_w(kw_j). \tag{6}$$

3.2.3 The first document selection

In our implementation, we select from the candidate document set D the document with the largest combination score of relevance and importance as the first document.

$$d^* = \text{argmax}_{d_i}(\lambda R(d_i, q) + (1 - \lambda)\alpha(d_i)), \forall d_i \in D \tag{7}$$

where $R(d_i, q)$ is the relevance score of document d_i to query q, which is calculated using Okapi BM25 score [17] in our implementation. $\alpha(d_i)$ is the importance score of document d_i, which is defined as the sum of the importance scores of the keywords d_i covers:

$$\alpha(d_i) = \sum_{j=1}^{m_d} (n_{kw_j}/n) \cdot \log_2(n/n_{kw_j}) \tag{8}$$

where n is the number of the candidate documents, m_d is the number of keywords in d_i, and n_{kw_j} is the number of the documents that contain keyword kw_j.

4. EXPERIMENTAL SETUP

4.1 The Test Collections

In our experiments, we utilized two public test collections to evaluate the effectiveness of KED.

The first test collection is called AMBIENT (AMBIgous ENTries)[1], which consists of 44 topics, each with a set of subtopics and a list of 100 ranked documents. The topics were selected from the list of ambiguous Wikipedia entries[2]. The documents were collected from a web search engine and subsequently were manually annotated with Wikipedia subtopic relevance judgments. Each associated document consists of URL, title, and snippet. More details about AMBIENT can be found in [18]. As the authors discussed in [18], the retrieved 100 documents for each topic didn't cover all the corresponding Wikipedia subtopics, while there were also many retrieved noticeable subtopics that were not present in the Wikipedia list. Hence, to make our evaluation more reasonable, we removed the documents which don't belong to any Wikipedia subtopic in our experiments.

The second test collection is a labeled query data set for TREC 6-8 Interactive Track. Since the original text of the documents is not public for free, we downloaded the transformed data set[3] used in [12], which consists of 17 files corresponding to the 17 queries. Each file contains all documents relevant to each query, and each document had been performed Porter stemming and stop-word removal. Candidate document sets contain on average 45 documents, 20 subtopics, and 300 words per document. Compared with the original TREC dataset, this public one ($TREC^{Yue}$ hereafter) means the same for our evaluation, except that Okapi method cannot be conducted and KED can only use all single words as keywords.

4.2 Evaluating Measures

4.2.1 Subtopic recall

Since we aim at providing diverse top results to satisfy different information needs, how many percents of the underlying facets covered by the top results is a natural evaluation measure. So we utilized subtopic recall [7] to evaluate the diversification performance.

Consider a topic T with n_t subtopics $ST_1, ST_2, \ldots, ST_{n_t}$, and a ranking $d_1, d_2, \ldots, d_n$ of n documents. Let $subtopic(d_i)$ be the set of subtopics to which d_i is relevant. Subtopic recall at rank k is defined as the percentage of subtopics covered by the first k documents, *i.e.*,

$$S\text{-}rec@k \equiv \frac{|\cup_{i=1}^{k} subtopic(d_i)|}{n_t}.$$

Intuitively, larger *S-rec* value means covering more subtopics. When all subtopics are covered, the value of *S-rec* equals to 1. For any topic, we can ideally find a minimal optimal rank so that the subtopic recall value at that rank is equal to 1. As users often expect to find their desired information in the first result page (usually the top 10 results), in our evaluation, we studied the performance upon average *S-rec* at rank 1, 5 and 10. Moreover, as different topics usually have different numbers of subtopics, to make evaluation more accurate, we utilized one more measure about subtopic recall, average *S-rec* at minimal rank (*avg_S-rec@minR*).

[1] http://credo.fub.it/ambient.
[2] http://en.wikipedia.org/wiki/Wikipedia:Links_to_%28disambiguation%29_pages.
[3] http://projects.yisongyue.com/svmdiv/.

4.2.2 Weighted subtopic loss

When different subtopics have different importance, such as popularity, the measures introduced in section 4.2.1 can't tell whether a diversification method prefers the more important subtopics. Hence, we utilized one more diversification measure, Weighted subtopic Loss [12]. Given a topic and its corresponding candidate document set, each subtopic's weight is proportional to the number of documents that cover the subtopic. Weighted Subtopic Loss at rank k is defined as the weighted percentage of subtopics not covered by the first k documents. In our evaluation, we utilized average Weighted Subtopic Loss at the minimal rank (*avg_WSL@minR*) overall the topics to measure the weighted diversification performance.

4.3 Experimental Settings

We compared our prototype with five existing implicit diversification approaches, *i.e.*, GRASSHOPPER [10] (G hereafter), MaxMinDispersion [8] (MMD hereafter), MaxSumDispersion [8] (MSD hereafter), Essential Pages [11] (EP hereafter) and SVM^{div} [12]. We also took a relevance-based search method Okapi [17] as a baseline in experiments on AMBIENT dataset.

For each document in AMBIENT, we extracted its title and snippet to construct a retrieved document. And then we did stemming and stop word removal on the retrieved documents. For each data file in $TREC^{Yue}$, we extracted the term id and term frequency values for each word.

We collected the top 15 ranked results of the methods, and the settings of the six methods are detailed as follows:

- For G, we represented documents as TFIDF vectors using all single words and constructed the similarity matrix using cosine values of the vectors. We gave a uniform distribution r and utilized the author's algorithm implementation[4].
- For MMD and MSD, we did the same vector construction as for G. Then, we utilized the TFIDF vectors to calculate the Euclidean Distance between the documents.
- For EP, we implemented the method according to the description in [11].
- For SVM^{div}, we trained the ranking model with the whole $TREC^{Yue}$ dataset utilized in [12] for evaluation on AMBIENT dataset. For evaluation on $TREC^{Yue}$ dataset, we used a 16/1 split for our training and test sets respectively. For training, we chose parameter $C = 0.1$. The author's algorithm implementation[5] was used.
- For Okapi, we chose the most popular parameter values, setting $K1 = 2.0$, $b = 0.75$.

5. RESULTS AND ANALYSIS

5.1 Diversification Performance

Table 1 shows the diversification performance of the seven methods on AMBIENT dataset. As a whole, KED gives the best overall performance, and the relevance-based method Okapi significantly underperforms other diversification methods.

[4] http://www.cs.wisc.edu/~jerryzhu/pub/grasshopper.m.
[5] http://projects.yisongyue.com/svmdiv/.

Table 1. The diversification performance on AMBIENT dataset.

Methods	*avg_S-rec*			*avg_S-rec@minR*	*avg_WSL@minR*
	@1	@5	@10		
Okapi	0.158	0.400	0.564	0.479	0.253
G	0.149	0.433	0.613	0.515	0.173
MMD	0.144	0.484	0.721	0.614	0.169
MSD	0.144	0.467	0.671	0.582	0.192
EP	0.144	0.484	0.683	0.603	0.138
SVM^{div}	**0.163**	0.475	0.660	0.571	0.148
KED	0.149	**0.553**	**0.776**	**0.684**	**0.100**

Specifically, at rank 1, SVM^{div} achieves the best average subtopic recall, though there is no significant difference among the performance of the methods. At rank 5, KED significantly outperforms other methods. MMD, EP, MSD and SVM^{div} play the second with very close performance. The first 10 results ranked by KED cover 77.6% subtopics on average, about 25% larger than what was achieved by G that performs the worst among the diversification methods.

Similarly with the performance of *avg_S-rec*@10, KED significantly outperforms other methods on *avg_S-rec@minR*, covering 68.4% subtopics at minimal rank on average, while Okapi performs the worst. G still achieves the lowest *avg_S-rec@minR* score among the diversification methods. We notice that though SVM^{div} doesn't perform well on *avg_S-rec@minR*, it achieves good performance on *avg_WSL@minR*, only underperforming KED and EP. This may be due to its ranking model that is trained for minimizing weighted subtopic loss. We can also see that the two dispersion methods, MMD and MSD, have significantly worse performance on *avg_WSL@minR* than other methods with close *avg_S-rec@minR* performance.

Table 2 shows the performance of the six methods on $TREC^{Yue}$ dataset. Generally speaking, KED_{sw}, SVM^{div} and G significantly outperform the other three methods.

Table 2. The diversification performance on $TREC^{Yue}$.

Methods	*avg_S-rec*			*avg_S-rec@minR*	*avg_WSL@minR*
	@1	@5	@10		
G	**0.128**	0.312	0.473	0.465	0.363
MMD	0.076	0.274	0.431	0.419	0.488
MSD	0.076	0.242	0.390	0.368	0.523
EP	0.082	0.298	0.438	0.420	0.462
SVM^{div}	0.111	0.330	**0.536**	0.478	**0.354**
KED_{sw}^{*}	0.112	**0.332**	0.523	**0.512**	0.357

* **KED_{sw}** represents KED using single words as keywords.

In particular, among the six methods, G performs best on *avg_S-rec* at rank 1 and scores nearly the sum of what MMD and MSD get. On the overall subtopic recall of the first page results, SVM^{div} and KED_{sw} significantly outperform other methods, covering 53.6% and 52.3% subtopics respectively.

KED_{sw} scores 0.512 on *avg_S-rec@minR*, significantly outperforming other methods,

while it gets 0.357 on *avg_WSL@minR*, slightly worse than the best performance achieved by SVM^{div}. This suggests that compared with SVM^{div}, KED_{sw} is not good enough for covering popular subtopics in diversification. SVM^{div} still performs better on *avg_WSL@minR* than *avg_S-rec@minR*. A big surprise to us is that G, which performs the worst among the diversification methods on AMBIENT dataset, performs closely to SVM^{div} both on *avg_S-rec@minR* and *avg_WSL@minR*.

In conclusion, KED effectively provides diverse results on the two different types of datasets, one with short search engine snippets and the other with long articles. The supervised learning based method SVM^{div} performs much better on the dataset from which it learns than the different kind of dataset they never learned before, and it significantly does well in what it learned for (*i.e.* weighted subtopic loss). Therefore, we can come to the conclusion that compared with the five existed implicit diversification methods, KED can stably achieve better diversification performance.

5.2 Further Exploration

5.2.1 Effect of keyword extraction

To study the effect of keyword extraction in our prototype, we conducted experiments on AMBIENT using KED without extracting keywords as we described in section 3.2.1. That means KED using all the single words as keywords. The experiment results are shown in Table 3. We can see that, the keyword extraction technique we utilized brings in about 10% improvement on both *avg_S-rec@minR* and *avg_WSL@minR* for KED. Moreover, without extracting keywords, KED still outperforms other methods.

Table 3. The diversification performance of the five heuristic methods with two different types of keywords. SW represents Single Word, and FSWP represents Frequent Single Words and Phrases.

Methods	*avg_S-rec@minR*		*avg_WSL@minR*	
	SW	FSWP	SW	FSWP
G	0.515	0.560	0.173	0.153
MMD	0.614	0.639	0.169	0.269
MSD	0.582	0.608	0.192	0.310
EP	0.603	0.572	0.138	0.148
KED	0.633	0.684	0.113	0.100

Furthermore, we also conducted experiments on other four heuristic diversification methods with extracted keywords. SVM^{div} is not involved as its learning features are not designed based on single words. The results are shown in Table 3. We notice that the method based on document similarity, G, benefits from the keyword extraction technique on both measures. This result matches the conclusion drawn in [19] that the suffix tree based keyword extraction can give a better measure for document similarity for web search result clustering. The two dispersion methods based on document distance make a small improvement on *avg_S-rec@minR* but a large deterioration on *avg_WSL@minR*. EP suffers from the extracted keywords.

5.2.2 Effect of keyword frequency threshold

This section illustrates the behavior of KED as we varied the Keyword Frequency Threshold *T* in our implementation of keyword extraction. Fig. 1 shows the performance of KED on AMBIENT dataset with *T* ranging from 2 to 8 upon both *avg_S-rec@minR* and *avg_WSL@minR*. KED achieves the highest *avg_S-rec@minR* when $T = 2$, though we note that there is no significant difference when *T* is not larger than 4. The performance decreases significantly after *T* increases to 5. Similar performance was achieved on *avg_WSL@minR*. When *T* is chosen larger than 4, KED achieves higher *avg_WSL@minR* as *T* grows.

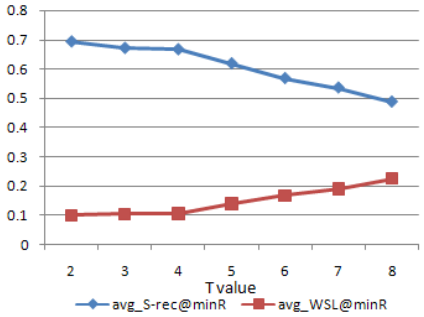

Fig. 1. Performance of KED with varying keyword frequency threshold *T*.

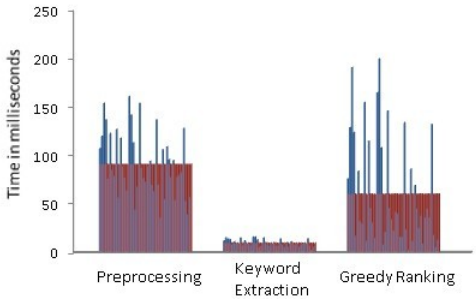

Fig. 2. Run-times for individual steps of our KED implementation. Results were obtained under the environment of Cygwin, using a 2.33GHz CPU with 4GB RAM.

5.2.3 Run-time analysis

Finally, we examine the run-time of our proposed method. Fig. 2 shows the actual run-times for the different processes involved in our implementation of KED on the AMBIENT dataset. We divide our implementation into three processes: preprocessing, keyword extraction and greedy ranking. From the figure, we can see that the run-times for different topics differ a lot due to the different numbers of their candidate documents. And the preprocessing step takes the longest run-time, 91 milliseconds on average; the greedy ranking step takes the second, taking average 60 milliseconds; the keyword extraction step takes

average 9.6 milliseconds. That means our implementation of KED takes less than 200 milliseconds to provide diverse results for each topic of the AMBIENT dataset, which is acceptable in practical applications.

5.3 A Query Example

To give a more specific demonstration of the diversification performance of KED, we took query 'Michael Jordan' as an example to compare KED against Google search engine. We downloaded the top 100 search results for the query from Google, and removed the image and video results. Each result contains title, snippet and URL. We represented each result with its title and snippet, and then re-ranked them using KED.

Table 4 shows the top 10 ranked results by Google and KED respectively. Among the top 10 results of Google, the top 7 results are all mainly about the player career of the famous basketball star, and the only one result about the famous machine learning researcher is ranked at the last one. To the contrary, KED provides only 3 results (*i.e.* 4, 5 and 9) that mainly introduce the basketball star. It also provides other facets of information about him, such as news, homepage on Facebook, his steakhouse, article commenting on him. Except the NBA star, KED also provides results about the researcher (*i.e.* 1), a caravan dealer named 'Michael Jordan Caravan' (*i.e.* 3). We can see that KED provides more diverse results in the first page. However, we also notice that, improving diversity may reduce the retrieval precision somehow. In this example, the second result from KED is mainly about NBA video and ticket share, not much relevant to the query.

Table 4. The top 10 ranked results by Google and KED.

rank	Google		KED	Google rank
1	http://en.wikipedia.org/wiki/Michael_Jordan		http://www.eecs.berkeley.edu/Faculty/Homepages/jordan.html	11
2	http://no.wikipedia.org/wiki/Michael_Jordan		http://www.michaeljordan.org/	50
3	http://www.nba.com/playerfile/michael_jordan/index.html		http://www.michaeljordancaravans.co.uk/	41
4	http://www.nba.com/history/players/jordan_summary.html		http://www.nba.com/history/players/jordan_summary.html	4
5	http://www.nba.com/jordan/		http://www.mjordan23.com/	33
6	http://www.nba.com/historical/playerfile/index.html?player=michael_jordan		http://www.washingtonpost.com/wp-dyn/content/article/2010/05/30/AR2010053003391.html	51
7	http://sports.espn.go.com/nba/players/profile%3fplayerid%3d1035		http://www.facebook.com/MichaelJordan	20
8	http://espn.go.com/sportscentury/features/00016048.html		http://www.nydailynews.com/topics/Michael+Jordan	35
9	http://search.espn.go.com/michael-jordan/		http://www.23jordan.com/	18
10	http://www.cs.berkeley.edu/~jordan/		http://www.michaeljordansteakhouse.com/	26

6. CONCLUSIONS

In this paper, we have proposed a novel prototype for web search result diversification. Given a set of candidate documents retrieved for a query, the *KEyword based Diversification* (KED) prototype provides a diverse ranking by extracting keywords from the retrieved documents to connect the documents and the underlying facets related to the query. The experiment results on two public datasets show that KED can stably outperform other existing implicit diversification approaches. Moreover, we have shown that its effectiveness can be further improved by using high quality keywords.

However, in this paper we only provide simple implementations of our prototype based on 'bag-of-words' model. There is so much room to improve the diversification performance of KED by using other implementations, for example, using other keyword extraction methods, other measures to describe the facet distance between keywords and so on. Furthermore, we can also improve the time effectiveness of our implementation and construct an on-line diversification system, like the existing web search result clustering systems.

REFERENCES

1. R. Agrawal, S. Gollapudi, A. Halverson, *et al*., "Diversifying search results," in *Proceedings of ACM International Conference on Web Search and Data Mining*, 2009, pp. 5-14.
2. R. L. T. Santos, J. Peng, C. Macdonald, *et al*., "Explicit search result diversification through sub-queries," in *Proceedings of European Conference on IR Research*, 2010, pp. 87-99.
3. W. Goffman, "On relevance as a measure," *Information Storage and Retrieval*, Vol. 2, 1964, pp. 201-203.
4. P. N. Bennett, B. Carterette, O. Chapelle, *et al*., "Beyond binary relevance: Preferences, diversity, and set-level judgments," *ACM SIGIR Forum*, Vol. 42, 2008, pp. 53-58.
5. F. Radlinski, B. Carterette, P. N. Bennett, *et al.*, "Redundancy, diversity and interdependent document relevance," *ACM SIGIR Forum*, Vol. 43, 2009, pp. 46-52.
6. J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, pp. 335-336.
7. C. X. Zhai, W. W. Cohen, and J. Lafferty, "Beyond independent relevance: methods and evaluation metrics for subtopic retrieval," in *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 10-17.
8. S. Gollapudi and A. Sharma, "An axiomatic approach for result diversification," in *Proceedings of the 18th International Conference on World Wide Web*, 2009, pp. 381-390.
9. S. Ravi, D. Rosenkrantz, and G. Tayi, "Facility dispersion problems: Heuristics and special cases," in *Proceedings of the 2nd Workshop on Algorithms and Data Structures*, 1991, pp. 355-366.

10. X. Zhu, A. B. Goldberg, J. Van, *et al*., "Improving diversity in ranking using absorbing random walks," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2007, pp. 97-104.
11. A. Swaminathan, C. Mathew, and D. Kirovski, "Essential pages," Technical Report, No. MSR-TR-2008-015, Microsoft Research, 2008.
12. Y. Yue and T. Joachims, "Predicting diverse subsets using structural SVMs," in *Proceedings of International Conference on Machine Learning*, 2008, pp. 1224-1231.
13. R. L. T. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proceedings of International Conference on World Wide Web*, 2010, pp. 881-890 .
14. S. Osinski and D. Weiss, "A concept-driven algorithm for clustering search results," *IEEE Intelligent Systems*, Vol. 20, 2005, pp. 48-54.
15. M. F. Porter, "An algorithm for suffix stripping," *Program*, Vol. 14, 1980, pp. 130-137.
16. D. Zhang and Y. Dong, "Semantic, hierarchical, online clustering of web search results," in *Proceedings of Asia-Pacific Web Conference*, 2004, pp. 69-78.
17. S. Robertson and K. S. Jones, "Simple proven approaches to text retrieval," Technical Report, No. TR356, Computer Laboratory, Cambridge University, 1997.
18. C. Carpineto, S. Mizzaro, G. Romano, *et al*., "Mobile information retrieval with search results clustering: Prototypes and evaluations," *Journal of American Society for Information Science and Technology*, Vol. 60, 2009, pp. 877-895.
19. H. Chim and X. Deng, "A new suffix tree similarity measure for document clustering," in *Proceedings of International Conference on World Wide Web*, 2007, pp. 121-130.

Gu-Li Lin (林古立) is a Ph.D. student of School of Computer Science and Engineering of South China University of Technology, Guangzhou, China. He received his B.Sc. degree in Computer Science and Technology from South China University of Technology, Guangzhou, China, in 2006. His research interests include information retrieval, web mining and machine learning.

Hong Peng (彭宏) received his Ph.D. degree in Mathematics from Xi'an Jiaotong University, Xi'an, China, in 1991, and finished his postdoctoral research at Zhejiang University, Hangzhou, China, in 1996. He is currently a Professor and Ph.D. supervisor of School of Computer Science and Engineering of South China University of Technology, Guangzhou, China. His areas of research includes data mining, machine learning and their applications.

Qian-Li Ma (馬千里) received his Ph.D. degree in Computer Science from South China University of Technology, Guangzhou, China, in 2008. He is now a lecturer of School of Computer Science and Engineering of South China University of Technology, Guangzhou, China. His research fields include nonlinear time series analysis and data mining.

Jia Wei (韋佳) received his Ph.D. degree in Computer Science from South China University of Technology, Guangzhou, China, in 2009. He is now a lecturer of School of Computer Science and Engineering of South China University of Technology, Guangzhou, China. His research fields include machine learning and images retrieval.

Jiang-Wei Qin (覃姜維) is a Ph.D. student of School of Computer Science and Engineering of South China University of Technology, Guangzhou, China. He received his B.Sc. degree in Computer Science and Technology from South China University of Technology, Guangzhou, China, in 2006. His research interests include transfer learning and text mining.