Improve Web Search Diversification with Intent Subtopic Mining[*]

Aymeric Damien, Min Zhang, Yiqun Liu, and Shaoping Ma

State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
aymeric.damien@gmail.com, {z-m,yiqunliu,msp}@tsinghua.edu.cn

Abstract. A number of search user behavior studies show that queries with unclear intents are commonly submitted to search engines. Result diversification is usually adopted to deal with those queries, in which search engine tries to trade-off some relevancy for some diversity to improve user experience. In this work, we aim to improve the performance of search results diversification by generating an intent subtopics list with fusion of multiple resources. We based our approach by thinking that to collect a large panel of intent subtopics, we should consider as well a wide range of resources from which to extract. The resources adopted cover a large panel of sources, such as external resources (Wikipedia, Google Keywords Generator, Google Insights, Search Engines query suggestion and completion), anchor texts, page snippets and more. We selected resources to cover both information seeker (What a user is searching for) and information provider (The websites) aspects. We also proposed an efficient Bayesian optimization approach to maximize resources selection performances, and a new technique to cluster subtopics based on the top results snippet information and Jaccard Similarity coefficient. Experiments based on TREC 2012 web track and NTCIR-10 intent task show that our framework can greatly improve diversity while keeping a good precision. The system developed with the proposed techniques also achieved the best English subtopic mining performance in NTCIR-10 intent task.

1 Introduction

Nowadays, most people are using Web search to look for information they need. Even if many of them will use targeted keywords, a great part will provide queries that can be interpreted in many different ways. So without further information about the user intent to disambiguate, search-engines have to focus on web search results diversification to produce a set of diversified results. There are two major kinds of query that needs to be diversified: ambiguous queries (e.g. "Jaguar"; that can be interpreted as

[*] This work was supported by Natural Science Foundation (60903107, 61073071) and National High Technology Research and Development (863) Program (2011AA01A205) of China.

G. Zhou et al. (Eds.): NLPCC 2013, CCIS 400, pp. 322–333, 2013.

the car brands, the animal, the Mac OS …) and broad queries (e.g. “Star Wars”; the user intent can be the movies, the video games, the books …). So for such queries, it is important that search-engines not only consider the relevancy of documents, but also provide a diversified set of documents that are covering different subtopics.

Other works proposed to solve this problem using different solutions, but most of these methods only rely on one or a few resources. As far as we know, this is the first time that resources fusion is explored to improve web search diversification and that a study is made about the different resources used to mine subtopics.

In this work, we first introduce the different resources used to extract subtopics and their processing methods. We applied two different process for candidates coming from external resources (such as Wikipedia, Google Keywords Generator, Search-Engines Suggestion …) and candidates coming from web pages (Retrieved through top results snippet information, anchor texts and page h tags of commercial search-engines or our own built search engine based on ClueWeb). For the first one, we introduce a new and efficient way to cluster the subtopics, based on the top results snippet information and Jaccard Similarity Coefficient. For the other one, we use the popular BM25 and Partition Around Medoid algorithms to cluster the subtopics.

We then propose a fusion of these resources in order to improve web search diversification. Furthermore we propose an interesting optimization to combine resources in order to maximize the performances of our framework.

Later, we give a complete analysis of our system performance using the Average Precision and D#-nDCG metrics, as well as a study of the good performances of the snippet based clustering method and a comparative review of the different resources used and their effectiveness by comparing different statistics.

Our work provide some new and original ways to improve web search results diversification:

- Fusion strategy of diverse and complementary resources are investigated to improve the results diversity while keeping an optimal relevancy.
- An optimization of the framework regarding the query type when selecting the resources to combine.
- A new subtopics clustering technique based on the top-results snippet information and Jaccard similarity coefficient is proposed and is evaluated to show its very good precision.
- Resources performance are comparatively studied to analyze each different resource effectiveness regarding the resource range of subtopic retrieved and relevancy, providing information about the best external resources to use in web search results diversification.
- Google Keywords Generator is used thought our work to generate a large list of subtopic candidates and get their popularity, and can be regarded as a really effective alternative source for finding query logs.

2 Related Work

Web search results diversification naturally appeared after the firsts page ranking algorithms in order to improve the search results ranking. Indeed, Zhai et al. [1] demonstrated that the relevancy could not only hold in a simple set of relevant results because of the correlations between these results, and that diversification were needed. Carbonell et al. [2] first introduced a model called re-ranking maximal marginal relevance (MMR) that not only focus on the relevance of the documents but also maximize the non-similarity between the results, in order to minimize the redundancy. As this algorithm does not imply classification of either the result or the query, diversification is leaded by the choice of similarity functions.

A later great concept was introduced by Ziegler et al. [3] about the diversification problem as a "recommendation" system. For each potential product result, the algorithm calculates the disparity between the item and all the other potential product result. Then it fusions this disparity to the original relevance order and then return the recommended products. Moreover they demonstrate experimentally that users were more likely to prefer more diversified results. This concept was then improved by Yu et al. [4] that proposed another recommendation system for results diversification by optimizing the balance between diversity and relevance. Indeed, the algorithm minimizes the correlation between documents by considering the constraint of relevance.

Agrawal et al. [5] introduced an approach of minimizing the risk of dissatisfaction of users, using taxonomy for classifying queries and documents, and creates a diverse set of results in accordance to it. The idea is that users only care about the top k returned result, and not the whole set.

Hu et al. [6] presented a method to understand the intent behind a user's query using Wikipedia. This system can help search engines to automatically route the query to some specific vertical search engines and obtain very relevant information. Their system is working by mapping the query into the Wikipedia intents representation space. They restricted their study to three different applications: travel, job, and person name.

Recently, Jiafeng et al. [7] also proposed an interesting method to measure query similarity with the awareness of potential search intents using a regularized topic model based on words from search result snippets and regularization from query co-clicks.

A part of our work uses similar techniques used by Han et al. [8] for subtopic mining from search-engines top results, using the BM25 and a partitioning around medoid clustering algorithm based on the cosine similarity. In this study, we adapted this method to fit our needs in the process of top-results pages subtopic extraction.

3 External Resources Based Subtopic Mining

Over the internet, there are many interesting services that we can use to help us to disambiguate a query. Indeed, from these resources, we can extract sub-intents of an ambiguous query as well as, for some, interesting information about the sub-intents popularity. In this part, we propose a new and efficient way to collect, filter and cluster all these sub-intents.

3.1 Resources Used

We decided to use a wide range of external resources, coming from commercial search-engines, such as query completion and suggestion, as well as specialized service providing popular query logs, such as Google Insights and Google Keywords Generator. At last, we also used Wikipedia encyclopedia for its disambiguation feature.

3.2 Subtopic Candidates Extraction and Filtering

Subtopics candidates had been extracted from different external resources: Query Suggestion and Completion from Google, Yahoo and Bing, Google Insights, Google Keywords Generator, and Wikipedia. For all, we can easily extract the candidates "as it". For example, Google Insights or Google Keywords Generator works like query logs, by giving the most popular keywords related to a query. So we submitted all our test queries to these resources and combine all the results data gathered using a linear combination.

Many data collected from these external resources are irrelevant or duplicated, that's why we applied a filter in order to keep only the valid ones. We applied a filter that remove all subtopics that do not contain all the query words, in any order. The original query stop words are discarded, so stop words does not need to be found in the candidates.

3.3 Snippet Based Clustering

Clustering has always been an important aspect in query diversification. The snippet information provided by each web page, bring us very relevant information that usually summarize the page, and list some important keywords. So we propose here a solution using this feature to cluster our subtopics. For each subtopic candidates, we first submit each one to the search engines (Google, Bing, Yahoo) and crawl the 50 top results snippet information. Then we set a table with every words found from these snippet and get their frequencies. After, in order to know if two subtopics are similar or not, we calculate the Jaccard similarity coefficient:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Where A and B are the two different term frequencies vectors of the two subtopics to compare. We extended this coefficient by considering both the words and their frequencies. (So even if many words retrieve for the two subtopics are the same, but their frequencies is really different, then their similarity will be reduced). We implemented this feature because we think that both words retrieved and their frequencies are important feature to know if two subtopics have similar intent or not. So when we calculate the intersection or the union of A and B, we added the average score of their frequencies:

$$J_{ext}(A,B) = \frac{\sum_{i \in A \cap B} \frac{f_{A_i} + f_{B_i}}{2}}{\sum_{i \in A \setminus B} f_{A_i} + \sum_{i \in B \setminus A} f_{B_i} + \sum_{i \in A \cap B} \frac{f_{A_i} + f_{B_i}}{2}} \tag{2}$$

With A and B the two frequencies vectors for every word and the frequency of i-th term of the vector. We then created a clustering algorithm using this extended Jaccard Similarity. The Jaccard similarity for a cluster is computed as the average similarity between all its subtopic candidates and all the other cluster subtopic candidates.

```
1. Select k (define experimentally)
2. Create for every subtopic candidate a cluster
3. For each cluster
   1. For each remaining cluster
          1. If Jext similarity of the the two clusters > k Then combine clusters
4. Repeat 3 while the similarity between two clusters is above k.
```

Alg 1. Bottom-up hierarchical clustering algorithm with extended Jaccard similarity coefficient

3.4 Resources Features Based Cluster Ranking

To rank our clusters, we based our approach on a multi criteria ranking. We used the different scores provided by the external resources; for example, in Google Insights or Google Keywords Generator, a score is associated with each term: the popularity for Google Insights and the amount of searches for Google Keywords Generator. So we applied a ranking based on the following features with their weight:

- Jaccard Similarity between the subtopic and the original query: 5%
- Google Insights score: 15%
- Google Keywords Generator score: 75%
- Belongs to the query suggestion/completion: 5%

We also considered that, if a subtopic belongs to the Wikipedia disambiguation feature, then the subtopic is important, and grants him a better score. In order to normalize all the scores to be able to compare them together, we convert each score into a percentage of the maximum score. So for example, the top search in Google insights or Google keywords generator will have a score of 1. Thanks to this normalization, even if the data come from different resources, we are still able to use them together.

4 Top Results Based Subtopic Mining

In this second approach, we propose to find the subtopics directly from the web pages. To get web pages related to the query to disambiguate, we used different search-engines: the commercial ones: Google, Bing and Yahoo, and the one built by THUIR (TMiner) that is based on the Clueweb data. In this way, we are sure to only extract pages that are very relevant to the query. We based our approach using a slightly similar method to the one proposed by Han et al. [8].

4.1 Subtopics Candidates Extraction

We first submitted the query to the search-engines, and get the page results. For TMiner run, we extracted the candidate subtopics from different fragments coming from page snippet, page h1 tags and in-link Anchors Text. For the commercial search-engines run, we only extracted the candidate subtopics from page snippet (page title and description). We adopted a vector space model to represent each fragment.

$$f = \left(w_{1,f}, w_{2,f}, \dots, w_{n,f}\right) \tag{3}$$

Where $w_{i,f}$ is the weight of a unique word i contained in f. We removed stop words and query words from the fragments, because they do not help us to distinguish the different fragments. We then used the BM25 [13] algorithm to evaluate the weight:

$$w_{i,f} = \frac{(k_1 + 1)tf_i}{k_1\left((1-b) + b\frac{dl}{avdl}\right) + tf_i} \log\frac{N - df_i + 0.5}{df_i + 0.5} \tag{4}$$

Where tf_{i} is the occurrence of word i in fragment f, and $\mathrm{d}f_{\mathrm{i}}$ is the number of documents that contains i in the corpus. *dl* is the length of the fragment f. *avdl* is the average fragment length for the query. N is the total number of documents in the entire corpus. We set k_1=1.1 and b=0.7.

4.2 Subtopic Candidates Clustering

We apply a modified Partitioning Around Medoids (PAM) clustering algorithm to group similar fragments together. Here is the algorithm:

```
1. Initialize: randomly select k of the n data points as the medoids
2. Associate each data point to the closest medoid. ("closest" here is defined using
   cosine similarity)
3. For each medoid m
4. For each non-medoid data point o
   1. Swap m and o and compute the total cost of the configuration
         1. Select the configuration with the lowest cost.
5. Repeat steps 2 to 4 until there is no change in the medoid.
```

Alg 2. Modified paritioning around medoid algorithm

The similarity between two fragments is determined using the cosine similarity between their corresponding weight vectors calculated as above using the BM25 algorithm. The PAM algorithm first computes k representative objects, called medoids. A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal. After finding the set of medoids, each object of the data set is assigned to the nearest medoid. k is the number of clusters we want to generate and traditionally it is fixed as an input of PAM. However, in our task, it is not suitable as we do not know the number of cluster (intent) a query has; indeed the number is not predictable. Those we had to modify the PAM algorithm to make it able to decide an appropriate k. We first randomly choose k points as initial cluster medoids. We then assign each other points to the closest medoid. If the closest of the medoid is over a value we set experimentally, then we set this point as a new medoid, and recalculate from the beginning.

4.3 Clusters Ranking

We rank the clusters according to their popularity; using the fragment rank inside the commercial search-engine or TMiner and the URLs diversity from the different fragments from a cluster. So we give a greater score to the clusters that contains fragments from higher ranked pages and clusters that contains fragments from many different URLs. Here is the formula used to calculate the score for each cluster:

$$Score(c) = \sum_{f \epsilon Frag(c)} 1 - \frac{w(f)}{N} \tag{5}$$

Where *w*(*f*) is the weight of the fragment, calculated by the fragments average position in the search results. Learning to rank techniques can also be adopted with sufficient training examples and we would like to add this to our future work.

4.4 Clusters Name Generation

From the different fragments, we need to generate a readable name for the cluster. We first select the most frequent word and then extend it to an n-gram based on the frequency of the other words. We also set that frequency limit experimentally. We kept stop words because they can be interesting to name the intent. Then we check if we need to add or not the keyword (that we removed from every fragment). So we compare if any original fragment contains or not the keyword, and if more than 50% contains it, we add it, using its position between the most frequent words, in order to place it correctly.

5 Resources Fusion

In order to improve the diversity, we combined the sub-intents we extracted from both external-resources and top-results pages mining. Indeed, both data are coming from two different aspect of the internet: one, from the external resources represents the queries that people are looking for on internet. And another one, the subtopic mining from the top-results pages, that shows the information provided by the website owners or participants.

A fusion is necessary over a unified model because the two sources of information are treated in two different ways; the external resources represent query logs keywords that we then cluster and rank, while in the top results mining, we do not directly cluster the subtopics, but we cluster sentence fragments, extracted from snippets, h1 tags or anchor texts. And then, we judge if a cluster is valid or not and generate a name to the intent. So the two approach are quite different and, in order to maximize the performances, a fusion approach is better than a unified model.

To combine our data, we used a linear combination of the sub-intents obtained from the external resources based mining and the top results based mining. After that combination, we expected that many sub-intents were actually duplicated, so we had to choose a way to recluster and rerank the sub-intents. We applied again our snippet-based clustering algorithm and experimentally define a similarity value to decide whether or not two candidates should be clustered together. At last, to rerank the data, we first normalized them by assigning them a percentage of the maximum sub-intent score for each query, so we could then compare the score of all sub-intents.

6 Fusion Optimization

Combining resources may not only result in performances increase, but may provoke a loss of relevancy or diversity. So we optimized the fusion process in order to get the best resources combination. So we ran some experiments to verify each resource performances compared to the fusion performance and we found out that the query type (navigational: Queries that aim to access a specific website or informational: Queries that aim to get general information about a topic) could impact the fusion. Indeed, we can see from figure 1 that navigational queries got worst performances after the fusion. So we optimized our system to take consider the query type and not apply the fusion, but only keep the candidate subtopics coming from the top results based mining method.

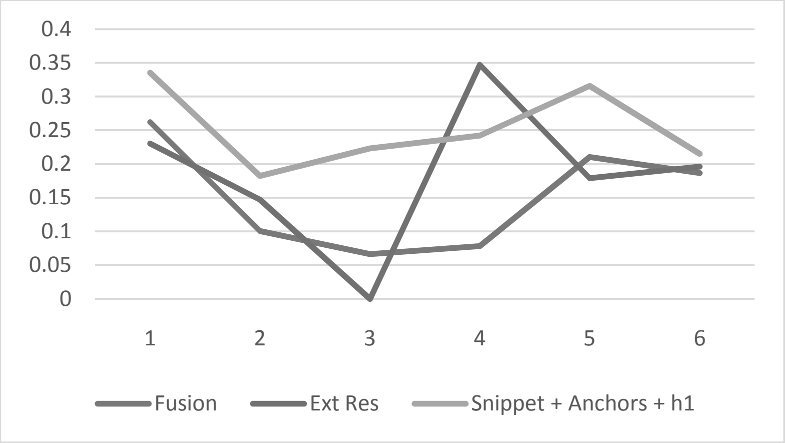

Fig. 1. Fusion Performances for Navigational Queries

Indeed, navigational queries aim to directly access a specific website, so subtopic extracted through the query logs like resources (like Google Key. Gen.) may not be that accurate and diverse. On the other hand, a webpage provide us a lot of possible intent subtopics, independently of the query type. So in the case of navigational queries, the mining from top results pages will outperform the one from external resources and the fusion.

7 Experiments and Discussions

7.1 Experiments Setup

Our experiment is based on a set of 50 queries, used for the TREC Web Track 2012. We tested our framework for each one of this queries and evaluated the system. To evaluate our work, we used several metrics and statistics, such as Average Precision metrics and D#-nDCG. Using these features, we aim to analyses three main part of our work: The efficiency of the snippet-based clustering, the different resources analysis, comparisons and performances and the entire method efficiency analysis for the web search results diversification.

D#-nDCG is a linear combination of intent recall (or “I-rec”, which measures diversity) and D-nDCG (which measures overall relevance across intents). The advantages of D#-nDCG over other diversity metrics such as α-nDCG [11] and Intent-Aware metrics [5] are discussed elsewhere [12]. We used the NTCIREVAL toolkit T. Sakai [10] to compute the above three metrics: thus, D#-nDCG is a simple average of I-rec and D-nDCG. We use the document cutoff of l = 10 throughout this paper, as a post hoc analysis of the runs showed that significance test results based on l = 20 and l = 30 are not so reliable.

7.2 Methods and Fusion Effectiveness

To test the fusion efficiency, we calculated the D#-nDCG of the fusion run, as well as the D#-nDCG for every one of its component (External Resources Mining and Top Results Mining). From the table 1, we can see that the I-rec value is higher for the fusion, meaning that the diversity is better. And on the other hand, the D-nDCG stay the same, implying that the subtopic relevancy has not been impacted by the fusion of the subtopics. Furthermore, the optimization run got even better performances, especially in term of relevancy, contributing to increase the framework efficiency in improving web search diversification.

The run with the Baseline and Wikipedia did not have much better performances due to the fact that few queries out of the 50 query set had a disambiguation topic in Wikipedia. Furthermore, the fusion did not get much better results than the Fusion because our 50 query set only contained 12% of navigational queries. But if we only analyze our system through these navigational queries, we can notice a raise of 23.40% in terms of D#-nDCG, with a raise of 40.14% in relevancy (D-nDCG). So we can conclude that this optimization really improve our system efficiency for navigational queries.

Table 1. Multi-resources fusion performances

Runs	D#-nDCG	I-rec	D-nDCG
Baseline (Query suggestion & completion)	0.23	0.2398	0.2203
Baseline + Wikipedia Disambiguation	0.2627	0.2735	0.2519
Baseline + Google Insights	0.3294	0.3116	0.3472
Baseline + Google Keywords Generator	0.367	0.3811	0.3529
Baseline + Google Keywords Generator + Google Insights + Wikipedia	0.3707	0.3908	0.3506
Baseline + TMiner Snippets, Anchor Texts and h1 Tags	0.3732	0.3971	0.3492
Baseline + Search-Engines & TMiner Snippets	0.3685	0.3809	0.3561
Baseline + Search Engines Snippets + TMiner Snippets, Anchor Texts and h1 tags	0.3787	0.4021	0.3553
Fusion (Baseline + Ext. Res. + SE & TMiner Snippets + TMiner h1 & Anchors)	0.4023	0.4542	0.3504
Fusion Optimization	0.4106	0.4587	0.3625

7.3 Resources Analysis and Comparisons

We can see from the table 2 that number of unique sub-intent coming from Google keywords generator is really higher than any other else resource. Then come the sub-intents extracted from the top-results pages titles, h1 tag and anchors. Finally, the other external resources like the query suggestion and completion, Google Insights and Wikipedia, brought relatively few unique queries. Indeed, these resources only brings the popular sub-intents, and not a large range of sub-intents. And we found that the percentage of unique sub-intents were quite similar for Google Key. Gen. and the

Table 2. Subtopic mining uniqueness performance of different resources of information

Resources	Unique Sub-intents	% of Unique Sub-intents
Google Key. Gen.	41.43	47.37%
TMiner Titles	13.00	43.07%
TMiner Anchors	12.50	46.23%
SE Titles	12.22	41.99%
TMiner H1	8.68	36.50%
Query Completion	3.12	5.57%
Query Suggestion	3.07	4.50%
Google Insights	2.75	3.17%
Wikipedia Disamb.	0.23	0.26%

subtopic extracted from top-results mining. Implying that combining data from both information seeker and information provider side has a great complementarity, and as a conclusion, improves the search result diversity. Moreover, even if some external resources could not provide some new intents, such as query suggestion or completion, we can still use them to evaluate the popularity of the sub-intent. So, these resources are still important to rank the subtopics.

7.4 Snippet-Based Clustering Performances

We found interesting to give some performance about the new clustering way we presented, based on the Jaccard Similarity of top results page snippets. We evaluated our technique by giving the average precision of the average percentage of clusters that first does not contains any subtopic incompatible with the other subtopics of that cluster. And then the percentage of clusters that does not have a duplicated intent with another cluster. At last, we deducted the percentage of clusters that are valid.

Table 3. Snippet based clustering precisions

	Average Precision
% clusters without wrongly added subtopics	0.953
% clusters with no duplicated intent	0.941
Total % of valid clusters	0.894

From our analysis, we can see that the precision for the two specific features of wrongly added subtopics and duplicated clusters are around the same and good, implying that the clustering performs well for both. As a result, the average precision of the clustering technique is very good by reaching 89.4%.

8 Conclusion and Future Work

To conclude, we can say that multi-resource fusion can greatly improve the web search result diversification. Indeed, considering both sub-intent mining from external services and web pages, enable us to have a wild range of different sub-intent sources. Indeed, we could explain it by the fact that, the external resources such as query suggestion or Google keywords generator, represent the information that the user is seeking, like query logs. While the web page, provide us another complementary range of topics, because they are information provided by websites that are expecting to fit the users information seeking needs. Furthermore, considering the query type to adapt the resources fusion let us improve our system and get even better results.

Then, we can observe that using query suggestion or completion and Google keywords generator external resources could be a good alternative for people seeking for query logs information. Indeed, no recent public query logs has been released by any commercial search-engine. So they could use these data instead. Moreover that the data has already been processed and some interesting features, like the amount of searches by month is provided.

We think that learning to rank techniques can also be adopted for our subtopics ranking in both method, and we would like to test different in order to improve the framework. At last, in this paper, we only focused on sub-intent extraction, clustering and ranking. In a future work, we would like to combine these data with document ranking and compare the results with some commercial search-engines results.

References

1. Zhai, C.X., Cohen, W.W., Lafferty, J.D.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR, pp. 10–17 (2003)
2. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, pp. 335–336 (1998)
3. Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: WWW 2005: Proceedings of the 14th International Conference on World Wide Web, pp. 22–32. ACM, New York (2005)
4. Yu, C., Lakshmanan, L., Amer-Yahia, S.: It takes variety to make a world: diversification in recommender systems. In: EDBT 2009: Proceedings of the 12th International Conference on Extending Database Technology, pp. 368–378. ACM, New York (2009)
5. Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: WSDM 2009: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 5–14. ACM, New York (2009)
6. Hu, J., Wang, G., Lochovsky, F., Tao Sun, J., Chen, Z.: Understanding user's query intent with Wikipedia. In: Proceedings of WWW 2009, pp. 471–480 (2009)
7. Guo, J., Cheng, X., Xu, G., Zhu, X.: Intent-aware query similarity. In: CIKM 2011, pp. 259–268 (2011)
8. Han, J., Wang, Q., Orii, N., Dou, Z., Sakai, T., Song, R.: Microsoft Research Asia at the NTCIR-9 Intent Task. In: NTCIR-9 Proceedings, pp. 116–122 (December 2011)
9. Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E., Milios, E.: Semantic similarity methods in wordNet and their application to information retrieval on the web. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 10–16 (2005)
10. Sakai, T.: NTCIREVAL: A generic toolkit for information access evaluation. In: Proceedings of FIT 2011, vol. 2, pp. 23–30 (2011)
11. Clarke, C.L.A., Craswell, N., Soboroff, I., Ashkan, A.: A comparative analysis of cascade measures for novelty and diversity. In: Proceedings of ACM WSDM 2011, vol. (2011)
12. Sakai, T., Song, R.: Evaluating Diversified Search ResultsUsing Per-Intent Graded Relevance. In: Proceedings of ACM SIGIR 2011, pp. 1043–1052 (2011)
13. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gatford, M., Payne, A.: Okapi at TREC-4. In: NIST Special Publication 500-236: The Fourth Text Retrieval Conference (TREC-4), pp. 73–96 (1995)