Enrichment and Reductionism: Two Approaches for Web Query Classification*

Ritesh Agrawal[1], Xiaofeng Yu[2], Irwin King[1,2], and Remi Zajac[1]

[1] AT&T Labs Research
201 Mission St. Ste 200
San Francisco, CA 94105
{ragrawal,irwin,remi}@research.att.com
http://www.research.att.com
[2] Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
{xfyu,king}@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk/~{xfyu,king}

Abstract. Classifying web queries into predefined target categories, also known as *web query classification*, is important to improve search relevance and online advertising. Web queries are however typically short, ambiguous and in constant flux. Moreover, target categories often lack standard taxonomies and precise semantic descriptions. These challenges make the web query classification task a non-trivial problem. In this paper, we present two complementary approaches for the web query classification task. First is the *enrichment method* that uses the World Wide Web (WWW) to enrich target categories and further models the web query classification as a search problem. Our second approach, the *reductionist approach*, works by reducing web queries to few central tokens. We evaluate the two approaches based on few thousands human labeled local and non-local web queries. From our study, we find the two approaches to be complementary to each other as the reductionist approach exhibits high precision but low recall, whereas the enrichment method exhibits high recall but low precision.

Keywords: Query Classification, Unsupervised, Semi-supervised, Bayesian Approach.

1 Introduction

With the increasing popularity of search engines as the de-facto gateway to the World Wide Web (WWW), web queries have become an important medium by which a system can understand user's interests. Web queries can be however very diverse and any meaningful use requires classifying them into small commercial

* This work was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413210).

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part III, LNCS 7064, pp. 148–157, 2011.

taxonomy. Several challenges however make the task web query classification a difficult and a non-trivial problem. Web queries are typically short containing mostly two or three terms [1]. As a result, they tend to be ambiguous. For instance, the term "apple" can be either mean the fruit or the company or a gadget (apple computers). Web queries are also in constant flux and keeps on changing with current ongoing events (such as release of Apple's IPad or Japan Earthquake, nuclear leak, etc.). Moreover, target categories are not fixed, they depend on business requirements and often lack precise clean semantic descriptions.

As highlighted by the 2005 KDD Cup Challenge[1], the above challenges associated with the web query classification task has generated lot of interest in the academia and in the industry. Building on existing research in this area, this paper presents two complementary approaches for the web query classification task. First, the *enrichment method* uses the World Wide Web (WWW) to enrich categories and models the web query classification as a search problem. Second, the *reductionist approach* reduces a query to smaller subset of tokens that maintain the broad intention of the query. This smaller set of tokens are referred as central terms of a query, hence the reductionist approach here is is sometimes referred as the *centroid approach*.

That paper is organized as follows. In Section 2, we first explore some of the relevant work and existing approaches for web query classification. Section 3 presents the underlying theory and implementation of the proposed approaches. Section 4 discusses our evaluation strategy; we use crowdsourcing to obtain human labeled queries for training and testing purpose and further use standard measures of precision, recall and $F1$ for evaluation. Lastly, Section 5 presents conclusions and a discussion on the complementary nature of the two approaches.

2 Related Work

The task of web query classification is to classify queries into a set of predefined categories. Unlike document classification techniques, web query classification techniques have to deal with short queries and lack rich set of textual features, required for the classification purpose. To overcome the lack of rich query features, many researchers proposed query-enrichment based methods [2,3], also called post-retrieval techniques. Query-enrichment associates a collection of text documents to every query by sending the query to a commercial search engine and collecting the search engine results. Each query is represented by a pseudo-document bundling together the titles and snippets of the top ranked search result pages. These pseudo-documents are then classified into the target categories using text classification techniques. Since the target categories typically does not have associated training data, the KDD CUP 2005 winning solution solved the training problem by using the Open Directory Project (ODP) to build a an ODP-based classifier. The ODP taxonomy is then mapped to the target categories using various methods [4]. Thus, the post-retrieval query document is

[1] `http://www.sigkdd.org/kdd2005/kddcup.html`

first classified into the ODP taxonomy, and the classifications are then mapped into the target categories for web query classification. Using the above approach, the KDD cup winning solution [5] achieved an F1 measure of 0.44, which shows that accurate and robust query classification is still a difficult and open research problem.

Broder et al. [3] avoid the need for mapping between taxonomies (for instance from ODP to target categories) by using a set of keywords, attached to categories by human editor, as training documents. Although, their method achieves very good results ($F1$=0.893) on tails queries, it is difficult to compare these results to 2005 KDD Cup results as they use very different target taxonomy and dataset. Another challenge with their approach is that often target categories are just labels without any description of keywords.

Beitzel et al. [6] exploits both labeled and unlabeled training data for this task. Diemert and Vandelle [7] propose an unsupervised method based on automatically built concept graphs for query categorization. Some work has been dedicated to using very large query logs as a source of unlabeled data to aid in automatic query classification. Wen et al. [8] proposed a clustering method for query classification, which tried to associate related queries by clustering session data of query logs. The session data contain multiple queries and click-through information from users. Wen et al. [8] considered terms from result documents that a set of queries has in common. The use of query keywords together with session data has shown to be effective for query clustering. Beitzel et al. [9] tried to exploit some association rules between query terms to help query classification. Furthermore, they exploited several classification approaches, emphasized on an approach adapted from computational linguistics named selectional preferences, and used unlabeled query log data to mine these rules and validate the effectiveness of their approaches.

3 Web Query Classification Approaches

As shown in Eq. (1), the web query classification task can be modeled as the challenge of finding a category (c_i) that has maximum probability given a web query (q) as,

$$QC = \arg\max_{c} P(c_i|q). \tag{1}$$

In this paper, we use two complementary approaches to compute the probability of a category (c_i) given a query (q). Each of the two approaches are discussed in greater details in the following subsections.

3.1 Enrichment Approach

As discussed in Section 2, many researchers use the query-enrichment approach to overcome the short nature of a web query. A query-enrichment method works by transforming web queries into a set of pseduo-documents extracted from the WWW. In our enrichment approach, we use similar process but enrich target

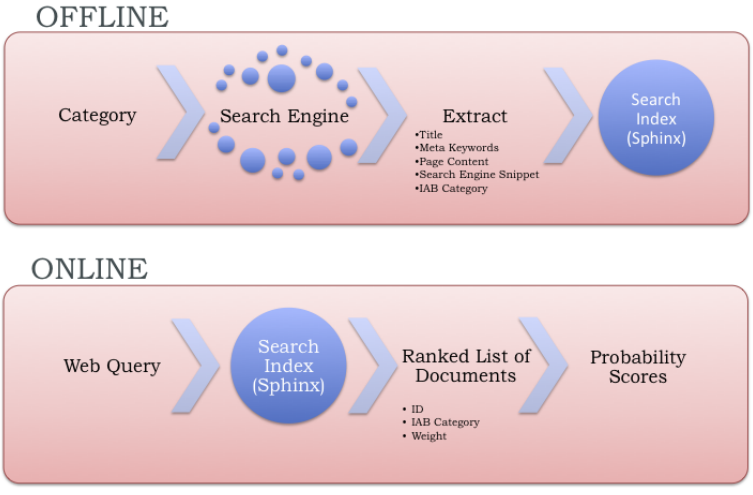

Fig. 1. Enrichment Method

categories instead of web queries. As shown in the Fig. 1, our category enrichment based approach is a two-step process.

Enriched Taxonomy. The first step, *an offline phase*, is the category enrichment process. Category enrichment is achieved by sending the category name as the search term to a commercial search engine and collecting the search engine results. For each category, we thus have a ranked list of documents where a document consists of a URL, a title, a search snippet, the URL's web page content, meta keywords and the category label that is used as a search term. The number of documents(γ) used per category is set empirically as described in Section 4.

Query Classification as Search. The second step, *an online phase*, is concerned with the actual web query classification task. Here we model the web query classification task as a search problem. Using Sphinx search engine[2], we first create a search index consisting of all the documents extracted in step 1. A web query that needs to be classified is then issued against this index. The results of this search includes a ranked list of indexed documents and associated BM25 relevance scores. Since, each document is assigned to one to more category from the offline phase, we can write the conditional probability of a category given a query as

$$P(c_i|q) = \sum_d P(c_i|d_j)P(d_j|q), \tag{2}$$

where $P(c_i|d_j)$ is the conditional probability of a category (c_i) given a document (d_j) and $P(d_j|q)$ is the conditional probability of a document given a web

[2] `http://sphinxsearch.com/`

query (q). $P(d_j|q)$ is calculated as the normalized BM25 score of a document. For $P(c_i|d_j)$, one can naively assume it to be binary depending upon whether a document belongs to a given category or nor or, as discussed below, use a Bayesian transformation as shown in Eq. (3).

$$P(c_i|d_j) = \frac{P(d_j|c_i)P(c_i)}{P(d_j)}. \tag{3}$$

We calculate prior category probability from human labelled training data. $P(d_j|c_i)$ and $P(d_j)$ are computed using category names as search phrases against the above built index.

Using Eqs. (2) and (3), the enrichment method returns probability scores for all target categories for a given web query. In order to select relevant categories, we introduce two hyper-parameters, (1) threshold (α) and (2) number of categories (β), that are empirically tuned by running several tests on the training dataset. The threshold (α) is the minimum acceptable probability score of a category required to qualify as a relevant category. For instance, a threshold value of 15% indicates that only categories for which probability score is more than 15% are relevant. In some cases, this might still lead to too many categories. For instance, it is possible to have about 20 categories if the threshold is set to a low 5%. Since typically a search query belongs to a few categories, we also set a hard limit (β) on the number of categories that are selected. Thus, in the above example, if β is set to 3 then only top 3 categories, when ordered by decreasing probability scores, are selected.

3.2 Reductionist Approach

The hypothesis of our reductionist approach (or the centroid approach) is that, if queries share equivalent or synonymous centroid terms, these queries are very likely to share the same categories. Here, the centroid term in a query is the term that represents the broad and major intention of this query. In other words, this approach aims at reducing query terms (or tokens) to a smaller set of centroid terms while maintaining the broad intention of queries. As an illustrative example, consider the two queries "*harvard university*" and "*the london college*". The centroid terms of these queries are "*university*" and "*college*", respectively. Suppose we know that "*harvard university*" belongs to the category "*education*", it is very likely that "*the london college*" also belongs to the same category. Since they share synonymous centroid terms "*university*" and "*college*". If we have a reasonably large size of (e.g., several thousands of) queries with labeled categories, we can use them for classifying new queries.

We explore crowdsourcing [10,11] to obtain the labeled queries. Crowdsourcing describes outsourcing of tasks to a large group of people instead of assigning such tasks to an in-house employee or contractor, and allows to complete standard tasks more accurately in less time and at lower cost. Using crowdsourcing platforms such as Amazon Mechanical Turk, we distribute our query labelling

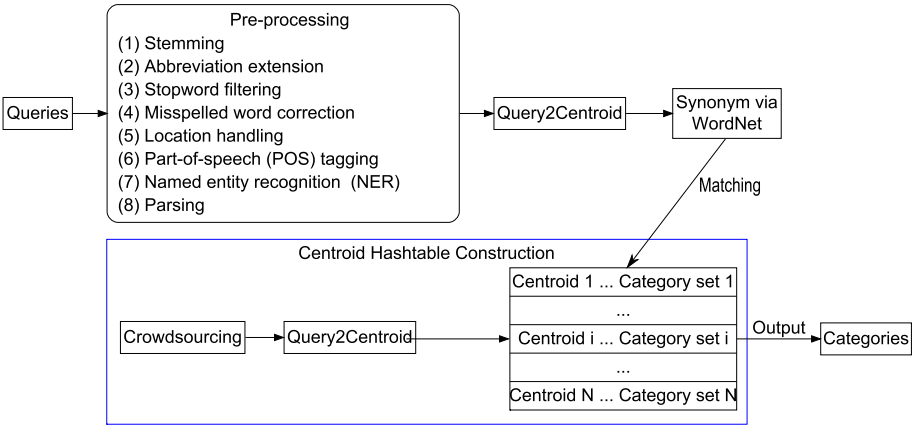

Fig. 2. Overall workflow for the centroid method

task to a large number of workers to obtain several labels per data point, and we apply statistical methods to filter out noisy labels. Consequently, we obtain approximately 3,300 labeled queries. We describe some details of our approach in Section 4.

Figure 2 shows the overall procedure. First, we perform some linguistics pre-processing of the query, including stemming, abbreviation extension, stopword filtering, misspelled word correction, location handling, part-of-speech (POS) tagging, named entity recognition (NER), etc. A number of off-the-shelf toolkits can be exploited for this purpose. For example, we use the porter stemming algorithm for query stemming, and we use Stanford POS tagger, NER tagger and parser for POS tagging, NER and parsing, respectively. The function Query2Centroid takes the pre-processed query as input, and identifies the centroid term of this query based on the POS tagging, NER and parsing results. For most queries, the centroid terms are nouns, verbs, noun or verb phrases. This considerably facilitates centroid identification, and we exploit rule-based methods for the Query2Centroid function, which performs effectively in practice. The function Synonym via WordNet returns all synonyms in WordNet for a query centroid term. For labeled queries, we conduct similar processing Query2Centroid to extract the centroid terms. We then construct a hash table in which the keys are centroid terms and the values are corresponding labeled categories. For an input query, we use synonyms of its centroid term to search the hash table to find the category. Note that the function Synonym via WordNet is useful since it enhances the coverage and boosts the recall. Take the query "*the london college*" for example to illustrate the procedure of the centroid method. After pre-processing, we list the POS tagging, NER and parsing results as follows:

POS tagging: `the/DT london/NNP college/NN`
NER: `the/O london/LOCATION college/O`
Parsing:

```
(ROOT
  (FRAG
    (NP (DT the) (NNP london) (NN college))))
det(college-3, the-1)
nn(college-3, London-2)
```

From these results, we know that "*london*" is a location name and college is a noun, and the function Query2Centroid can easily extract the centroid term "*college*" for the query "*the london college*". The function Synonym via WordNet finds all synonyms of "*college*" as "*college*" and "*university*". Suppose the hash table contains the key "*university*" and corresponding value (category) "*education*", we can easily and quickly find the category "*education*" for the query "*the london college*".

In summary, the centroid method is quite easy to implement as compared for example to related approaches mentioned in Section 2, while also performing reasonably well. The hash table look-up is very fast in on-line scenarios. This approach has a few limitations. First, as a reductionist approach, this method loses some information for query representation. Second, the quality of centroid identification (e.g., incorrect centroid terms) will affect the categorization performance. Third, for some queries, the categories cannot be found in the hash table. Therefore, this method exhibits high precision but low recall.

4 Experiment

Preparing Testing and Training Dataset. In order to test the two approaches, we extracted several thousands local and non-local web queries from search logs of two different commercially available search engines. A local search (such as "pizza near glendale ca", "walmart in new jersey city", etc.) is a specialization of the web (or non-local) search that allows users to submit geographical contained queries [12]. A non-lcoal search is any web search and ranges navigational searches (such as facebook, amzon api, etc) to informational searches (such as major stars in solar system, effects of global warming, etc.). Using Amazon Mechnical Turk's crowdsourcing system, each sampled query was labeled by 10 workers and assigned to one of the target category by each worker. In order to filter noisy data in the collected labelled data, we use two filtering steps. First, each worker is presented with a golden set of test queries randomly inserted into the labelling task. A golden set test query is a query for which we can easily and in an unambiguous manner identify the right category. For instance, one can easily say that "*Italian restaurant*" belongs to "*Food & Drink*" category given that there is no other category either related to Italian and restaurants. For each worker, we then calculate percentage adherence to these golden set queries by matching their selected categories to the expected target categories. We ignore all the labels from a worker who tends to differ from the golden set by more than 70%. This step removed about 3% labeled data.

Table 1. Performance of Enrichment and Centroid Method

	Local Search		**Non-Local Search**	
	Enrichment	**Centroid**	**Enrichment**	**Centroid**
Precision	0.613	0.616	0.428	0.707
Recall	0.421	0.409	0.281	0.233
$F1$	0.499	0.491	0.339	0.350

Second, for each web query, we calculate number of votes received across all the target categories and select only those categories that received more than 35% of votes. For instance, assume that out of 10 worker 4 say that the query "*financial news*" belongs to "*news*" category, while the other 4 say that it belongs to "*finance*" category and 2 say that it belongs to "*business*" category. In this particular, case we then accept "*news*" and "*finance*" as two correct categories and reject the "*business*" category. However, this constraint also introduces a limitation in our experiment. By restricting to only those categories that received more than 35% of votes, we are restricted to at-most two categories per query.

After this filtering, we are finally left with about 3,353 local and 3,324 non-local human labeled queries. Only 5% of queries in both the datasets have two categories associated with them. We randomly select 20% queries in the two datasets for testing purpose and use the rest 80% for training.

Results. As discussed in Section 3.1, the enrichment approach has three hyper-parameters, namely (1) threshold (α), (2) number of categories (β), and (3) number of documents (γ) used to represent a target category. In order to select optimal parameters, we run the enrichment algorithm several times on the training data with different parameter values. Figure 3 shows the influence of the three hyper-parameters on the $F1$ measure. In each of the figure, one parameter is varied while keeping the other two parameters constant. In Fig. 3 one notices the optimal performance of the enrichment method occurs by setting number of documents to 300, threshold to 15% and considering top two categories. We use the following settings to evaluate the performance of the enrichment method on the testing data.

Table 1 shows the precision, recall, and $F1$ measures for the two approaches on the local and non-local search queries. One notices that the performance of both the approaches (based on $F1$ Measure) significantly decreases for non-local searches as compared to local searches. This is partly because non-local searches tend to be much more diverse. For instance, a quick analysis of one month of local and non-local search log indicates that there are only 5% unique local searches in contrast to 23% unique non-local searches. This indicates that, as compared to local searches, non-local searches tend to be much more diverse and hence explains decreased performance of the two approaches on non-local searches.

Additionally, from Table 1 one also notices that the two approaches have comparable $F1$ measure; however, they demonstrate very different behavior. The centroid method exhibits higher precision as compared to the enrichment method, whereas the enrichment method shows higher recall as compared to the

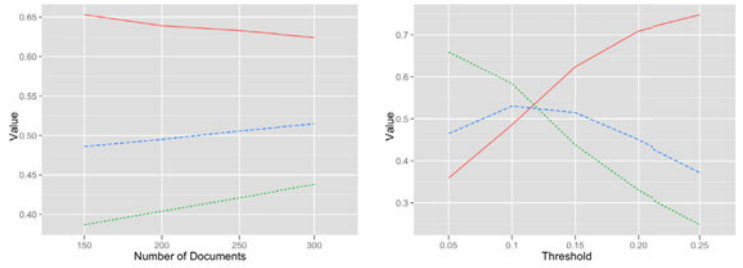

Fig. 3. Influence of hyper-parameters on the performance of the Enrichment method

centroid method. This is expected as the centroid method uses much more precise data (labeled queries) as compared to the enrichment data (online resources). As a result, the centroid method displays higher precision as compared to the enrichment method. On the other hand, however, using only labeled queries in part restricts the ability of the centroid method to deal with unseen centroid terms. In contrast, the noise in online documents helps improve recall of the enrichment method.

5 Conclusion

In this paper, we present two approaches–enrichment and reductionism, for the web query classification purpose. As demonstrated from the experiment, the two approaches are complementary in many different ways. First, on the theoretical level, the two methods approach the web query classification problem from two different ends. While the enrichment method focuses on enriching target categories, the reductionist approach focuses on reducing web queries to a few centroid terms. Second, at the pragmatic level, the two approaches demonstrate complementary precision and recall. The enrichment method has high recall but

low precision, while the centroid method has high precision but low recall. The complementary nature of the two approaches indicates a much higher performance can be achieved by combining the two using an ensemble technique [13], an aspect that we aim to further explore in future.

References

1. Baeza-Yates, R., Castillo, C.: Relating web structure and user search behavior. In: 10th World Wide Web Conference, Hong Kong, pp. 1–2 (2001)
2. Shen, D., Pan, R., Sun, J.-T., Pan, J.J., Wu, K., Yin, J., Yang, Q.: Query enrichment for web-query classification. ACM Trans. Inf. Syst. 24, 320–352 (2006)
3. Broder, A.Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., Zhang, T.: Robust classification of rare queries using Web knowledge. In: Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, pp. 231–238 (2007)
4. Shen, D., Sun, J.T., Yang, Q., Chen, Z.: Building bridges for Web query classification. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, pp. 131–138 (2006)
5. Shen, D., Pan, R., Sun, J.-T., Pan, J.J., Wu, K., Yin, J., Yang, Q.: Q2c@ust: our winning solution to query classification in kddcup 2005. ACM SIGKDD Explorations Newsletter 7, 100–110 (2005)
6. Beitzel, S.M., Jensen, E.C., Frieder, O., Grossman, D., Lewis, D.D., Chowdhury, A., Kolcz, A.: Automatic Web query classification using labeled and unlabeled training data. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, pp. 581–582 (2005)
7. Diemert, E., Vandelle, G.: Unsupervised query categorization using automatically-built concept graphs. In: Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, pp. 461–470 (2009)
8. Wen, J.-R., Jian-Yun Nie, H.J.Z.: Query clustering using user logs. ACM Trans. Inf. Syst. 20(1), 59–81 (2002)
9. Beitzel, S.M., Jensen, E.C., Lewis, D.D., Chowdhury, A., Frieder, O.: Automatic classification of Web queries using very large unlabeled query logs. ACM Trans. Inf. 25(2) (April 2007)
10. Howe, J.: The rise of crowdsourcing. Wired 14(6) (2006)
11. Alonso, O., Lease, M.: Crowdsourcing 101: Putting the "wisdom of the crowd" to work for you. In: Tutorials of the 4th ACM International Conference on Web Search and Data Mining (2011)
12. Agrawal, R.J., Shanahan, J.G.: Location disambiguation in local searches using gradient boosted decision trees. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2010, pp. 129–136. ACM, San Jose (2010)
13. Dzeroski, S., Zenko, B.: Is combining classifiers better than selecting the best one. In: Proceedings of the Nineteenth International Conference on Machine Learning, ICML 2002, pp. 123–130. Morgan Kaufmann Publishers Inc., San Francisco (2002)