Learning Semantic Categories from Clickthrough Logs

Mamoru Komachi
Nara Institute of Science and Technology (NAIST)
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
mamoru-k@is.naist.jp

Shimpei Makimoto and **Kei Uchiumi** and **Manabu Sassano**
Yahoo Japan Corporation
Midtown Tower, 9-7-1 Akasaka, Minato-ku, Tokyo 107-6211, Japan
{smakimot,kuchiumi,msassano}@yahoo-corp.jp

Abstract

As the web grows larger, knowledge acquisition from the web has gained increasing attention. In this paper, we propose using web search clickthrough logs to learn semantic categories. Experimental results show that the proposed method greatly outperforms previous work using only web search query logs.

1 Introduction

Compared to other text resources, search queries more directly reflect search users' interests (Silverstein et al., 1998). Web search logs are getting a lot more attention lately as a source of information for applications such as targeted advertisement and query suggestion.

However, it may not be appropriate to use queries themselves because query strings are often too heterogeneous or inspecific to characterize the interests of the user population. Although it is not clear that query logs are the best source of learning semantic categories, all the previous studies using web search logs rely on web search query logs.

Therefore, we propose to use web search clickthrough logs to learn semantic categories. Joachims (2002) developed a method that utilizes clickthrough logs for training ranking of search engines. A *search clickthrough* is a link which search users click when they see the result of their search. The intentions of two distinct search queries are likely to be similar, if not identical, when they have the same clickthrough. Search clickthrough logs are thus potentially useful for learnin semantic categories. Clickthrough logs have the additional advantage that they are available in abundance and can be stored at very low cost.[1] Our proposed method employs search clickthrough logs to improve semantic category acquisition in both precision and recall.

We cast semantic category acquisition from search logs as the task of learning labeled instances from few labeled seeds. To our knowledge this is the first study that exploits search clickthrough logs for semantic category learning.[2]

2 Related Work

There are many techniques that have been developed to help elicit knowledge from query logs. These algorithms use contextual patterns to extract a category or a relation in order to learn a target *instance* which belongs to the category (e.g. *cat* in *animal* class) or a pair of words in specific relation (e.g. *headquarter* to a *company*). In this work, we focus on extracting named entities of the same class to learn semantic categories.

Paşca and Durme (2007) were the first to discover the importance of search query logs in natural language processing applications. They focused on learning attributes of named entities, and thus their objective is different from ours. Another line of new research is to combine various resources such as web documents with search query logs (Paşca and Durme, 2008; Talukdar et al., 2008). We differ from this work in that we use search clickthrough logs rather than search query logs.

Komachi and Suzuki (2008) proposed a bootstrapping algorithm called *Tchai*, dedicated to the task of semantic category acquisition from search query logs. It achieves state-of-the-art performance for this task, but it only uses web search query logs.

[1] As for data availability, MSN Search query logs (RFP 2006 dataset) were provided to WSCD09: Workshop on Web Search Click Data 2009 participants. http://research.microsoft.com/en-US/um/people/nickcr/WSCD09/

[2] After the submission of this paper, we found that (Xu et al., 2009) also applies search clickthrough logs to this task. This work independently confirms the effectiveness of clickthrough logs to this task using different sources.

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 189–192,
Suntec, Singapore, 4 August 2009.

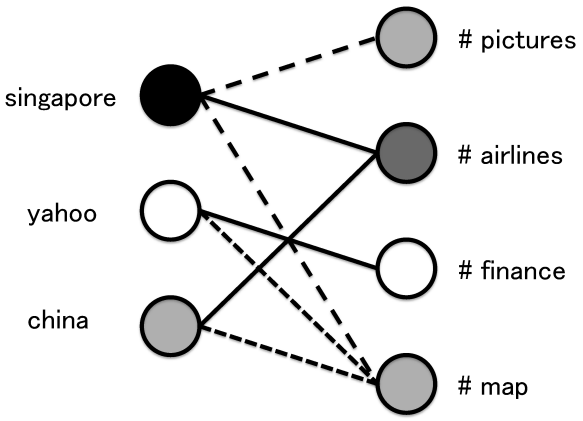

Figure 1: Labels of seeds are propagated to unlabeled nodes.

3 *Quetchup*[3] Algorithm

In this section, we describe an algorithm for learning semantic categories from search logs using label propagation. We name the algorithm *Quetchup*.

3.1 Semi-supervised Learning by Laplacian Label Propagation

Graph-based semi-supervised methods such as label propagation are known to achieve high performance with only a few seeds and have the advantage of scalability.

Figure 1 illustrates the process of label propagation using a seed term "singapore" to learn the Travel domain.

This is a bipartite graph whose left-hand side nodes are terms and right-hand side nodes are patterns. The strength of lines indicates relatedness between each node. The darker a node, the more likely it belongs to the Travel domain. Starting from "singapore," the pattern "♯ airlines" [4] is strongly related to "singapore," and thus the label of "singapore" will be propagated to the pattern. On the other hand, the pattern "♯ map" is a neutral pattern which co-occurs with terms other than the Travel domain such as "google" and "yahoo." Since the term "china" shares two patterns, "♯ airlines" and "♯ map," with "singapore," the label of the seed term "singapore" propagates to "china." "China" will then be classified in the Travel domain. In this way, label propagation gradually propagates the label of seed instances to neighbouring nodes, and optimal labels are given as the labels at which the label propagation process has converged.

Input:
Seed instance vector $F(0)$
Instance similarity matrix A
Output:
Instance score vector $F(t)$
1: Construct the normalized Laplacian matrix $L = I - D^{-1/2}AD^{-1/2}$
2: Iterate $F(t+1) = \alpha(-L)F(t) + (1-\alpha)F(0)$ until convergence

Figure 2: Laplacian label propagation algorithm

Figure 2 describes label propagation based on the *regularized Laplacian*. Let a sample x_i be $x_i \in \mathcal{X}$, $F(0)$ be a score vector of x comprised of a label set $y_i \in \mathcal{Y}$, and $F(t)$ be a score vector of x after step t. *Instance-instance similarity matrix* A is defined as $A = W^T W$ where $W$is a row-normalized *instance-pattern matrix*. The (i, j)-th element of W_{ij} contains the normalized frequency of co-occurrence of instance x_i and pattern p_j. D is a diagonal degree matrix of N where the (i, i)th element of D is given as $D_{ii} = \sum_j N_{ij}$.

This algorithm in Figure 2 is similar to (Zhou et al., 2004) except for the method of constructing A and the use of graph Laplacian. Zhou et al. proposed a heuristic to set $A_{ii} = 0$ to avoid self-reinforcement[5] because Gaussian kernel was used to create A. The Laplacian label propagation does not need such a heuristic because the graph Laplacian automatically reduces self-reinforcement by assigning negative weights to self-loops.

In the task of learning one category, scores of labeled (seed) instances are set to 1 whereas scores of unlabeled instances are set to 0. The output is a score vector which holds relatedness to seed instances in descending order. In the task of learning two categories, scores of seed instances are set to either 1 or -1, respectively, and the final label of instance x_i will be determined by the sign of output score vector y_i.

Label propagation has a parameter $\alpha \in (0, 1]$ that controls how much the labels of seeds are emphasized. As α approaches 0 it puts more weight on labeled instances, while as α increases it employs both labeled and unlabeled data.

There exists a closed-form solution for Laplacian label propagation:

[3] Query Term Chunk Processor

[4] ♯ is the place into which a query fits.

[5] Avoiding self-reinforcement is important because it causes semantic drift, a phenomenon where frequent instances and patterns unrelated to seed instances infect semantic category acquisition as iteration proceeds.

Category	Seed
Travel	jal (Japan Airlines), ana (All Nippon Airways), jr (Japan Railways), じゃらん (jalan: online travel guide site), his (H.I.S.Co.,Ltd.: travel agency)
Finance	みずほ銀行 (Mizuho Bank), 三井住友銀行 (Sumitomo Mitsui Banking Corporation), jcb, 新生銀行 (Shinsei Bank), 野村證券 (Nomura Securities)

Table 1: Seed terms for each category

$$F^* = \sum_{t=0}^{\infty}(\alpha(-L))^t F(0) = (I + \alpha L)^{-1} F(0)$$

However, the matrix inversion leads to $O(n^3)$ complexity, which is far from realistic in a real-world configuration. Nonetheless, it can be approximated by fixing the number of steps for label propagation.

4 Experiments with Web Search Logs

We will describe experimental result comparing a previous method *Tchai* to the proposed method *Quetchup* with clickthrough logs (*Quetchup*$_{click}$) and with query logs (*Quetchup*$_{query}$).

4.1 Experimental Settings

Search logs We used Japanese search logs collected in August 2008 from Yahoo! JAPAN Web Search. We thresholded both search query and clickthrough logs and retained the top 1 million distinct queries. Search logs are accompanied by their frequencies within the logs.

Construction of an instance-pattern matrix We used clicked links as clickthrough patterns. Links clicked less than 200 times were removed. After that, links which had only one co-occurring query were pruned. [6] On the other hand, we used two term queries as contextual patterns. For instance, if one has the term "singapore" and the query "singapore airlines," the contextual pattern "♯ airlines" will be created. Query patterns appearing less than 100 times were discarded.

The (i, j)-th element of a row-normalized instance-pattern matrix W is given by $W_{ij} = \frac{|x_i, p_j|}{\sum_k |x_i, p_k|}$.

Target categories We used two categories, Travel and Finance, to compare proposed methods with (Komachi and Suzuki, 2008).

When a query was a variant of a term or contains spelling mistakes, we estimated original form and manually assigned a semantic category. We allowed a query to have more than two categories. When a query had more than two terms, we assigned a semantic category to the whole query taking each term into account.[7]

System We used the same seeds presented in Table 1 for both *Tchai* and *Quetchup*. We used the same parameter for *Tchai* described in (Komachi and Suzuki, 2008), and collected 100 instances by iterating 10 times and extracting 10 instances per iteration. The number of iteration of *Quetchup* is set to 10. The parameter α is set to 0.0001.

Evaluation It is difficult in general to define recall for the task of semantic category acquisition since the true set of instances is not known. Thus, we evaluated all systems using *precision at* k and *relative recall* (Pantel and Ravichandran, 2004).[8] Relative recall is the coverage of a system given another system as baseline.

4.2 Experimental Result

4.2.1 Effectiveness of Clickthrough Logs

Figures 3 to 6 plot precision and relative recall for three systems to show effectiveness of search clickthrough logs in improvement of precision and relative recall. Relative recall of *Quetchup*$_{click}$ and *Tchai* were calculated against *Quetchup*$_{query}$.

Quetchup$_{click}$ gave the best precision among three systems, and did not degenerate going down through the list. In addition, it was demonstrated that *Quetchup*$_{click}$ gives high recall. This result shows that search clickthrough logs effectively improve both precision and recall for the task of semantic category acquisition.

On the other hand, *Quetchup*$_{query}$ degraded in precision as its rank increased. Manual check of the extracted queries revealed that the most prominent queries were Pornographic queries, followed by Food, Job and Housing, which frequently appear in web search logs. Other co-occurrence metrics such as pointwise mutual information would be explored in the future to suppress the effect of frequent queries.

In addition, *Quetchup*$_{click}$ constantly outperformed *Tchai* in both the Travel and Fi-

[6] Pruning facilitates the computation time and reduces the size of instance-pattern matrix drastically.

[7] Since web search query logs contain many spelling mistakes, we experimented in a realistic configuration.

[8] Typically, precision at k is the most important measure since the top k highest scored terms are evaluated by hand.

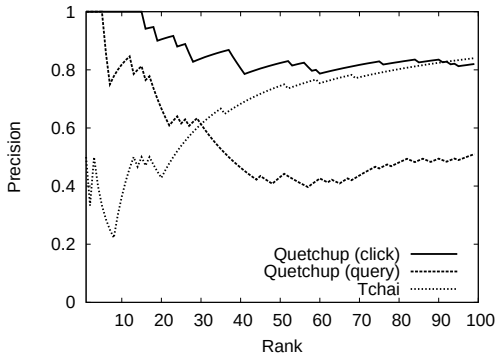

Figure 3: Precision of Travel domain

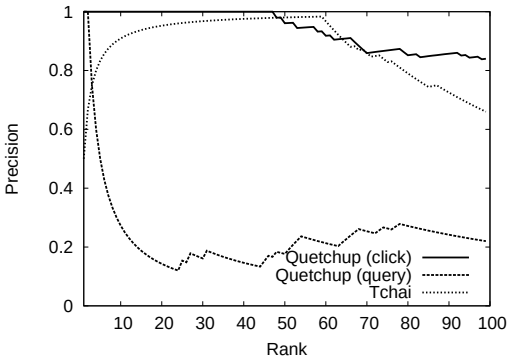

Figure 4: Precision of Finance domain

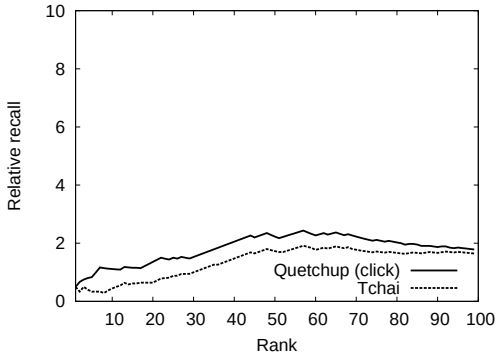

Figure 5: Relative recall of Travel domain

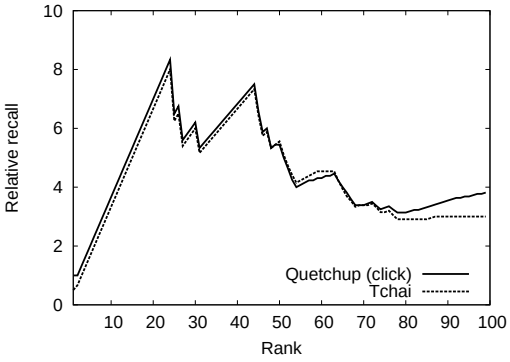

Figure 6: Relative recall of Finance domain

nance domains in precision and outperfomed *Quetchup*$_{query}$ in relative recall. The differences between the two domains of query-based systems seem to lie in the size of correct instances. The Finance domain is a closed set which has only a few effective query patterns, whereas Travel domain is an open set which has many query patterns that match correct instances. *Quetchup*$_{click}$ has an additional advantage that it is stable across over the ranked list, because the variance of the number of clicked links is small thanks to the nature of the ranking algorithm of search engines.

5 Conclusion

We have proposed a method called *Quetchup* to learn semantic categories from search clickthrough logs using Laplacian label propagation. The proposed method greatly outperforms previous method, taking the advantage of search clickthrough logs.

Acknowledgements

The first author is partly supported by the grant-in-aid JSPS Fellowship for Young Researchers. We thank the anonymous reviewers for helpful comments and suggestions.

References

T. Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. *KDD*, pages 133–142.

M. Komachi and H. Suzuki. 2008. Minimally Supervised Learning of Semantic Knowledge from Query Logs. *IJCNLP*, pages 358–365.

M. Paşca and B. V. Durme. 2007. What You Seek is What You Get: Extraction of Class Attributes from Query Logs. *IJCAI-07*, pages 2832–2837.

M. Paşca and B. V. Durme. 2008. Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. *ACL-2008*, pages 19–27.

P. Pantel and D. Ravichandran. 2004. Automatically Labeling Semantic Classes. *HLT/NAACL-04*, pages 321–328.

C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. 1998. *Analysis of a Very Large AltaVista Query Log*. Digital SRC Technical Note 1998-014.

P. P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira. 2008. Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks. *EMNLP-2008*, pages 581–589.

G. Xu, S. Yang, and H. Li. 2009. Named Entity Mining from Click-Through Log Using Weakly Supervised Latent Dirichlet Allocation. *KDD*. to appear.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schökopf. 2004. Learning with Local and Global Consistency. *NIPS*, 16:321–328.