Japanese Abbreviation Expansion with Query and Clickthrough Logs

Kei Uchiumi[†] **Mamoru Komachi**[‡] **Keigo Machinaga**[†]

Toshiyuki Maezawa[†] **Toshinori Satou**[†] **Yoshinori Kobayashi**[† *]

†**Yahoo Japan Corporation**
Midtown Tower, 9-7-1 Akasaka, Minato-ku, Tokyo 107-6211, Japan
{kuchiumi,kmachina,tmaezawa,toshsato}@yahoo-corp.jp
ykobayas@google.com

‡**Nara Institute of Science and Technology**
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
komachi@is.naist.jp

Abstract

A novel reranking method has been developed to refine web search queries. A label propagation algorithm was applied on a clickthrough graph, and the candidates were reranked using a query language model. Our method first enumerates query candidates with common landing pages with regard to the given query to create a clickthrough graph. Second, it calculates the likelihood of the candidates, using a language model generated from web search query logs. Finally, the candidates are sorted by the score calculated from the likelihood and label propagation. As a result, high precision and coverage were achieved in the task of Japanese abbreviation expansion, without using handcrafted training data.

1 Introduction

The query expansion technique has been widely used in recent web-search engines. Query expansion significantly improves recall in information retrieval operations. It uses a thesaurus or synonym dictionary to reformulate a query, or to correct spelling errors in search queries.

In the early days of the speller, the dictionary was manually compiled by lexicographers. However, it is time consuming to construct a broad coverage dictionary, and domain knowledge is required to achieve high quality. Moreover, the rapid growth of the web makes it even harder to maintain an up-to-date dictionary for the web.

To alleviate this problem, web-search engines often exploit web search query logs to automatically generate a thesaurus. A web search query is a query that a web user types into a web search engine to find information. It is noisy and sometimes ambiguous to detect query intent, but it is a great way to create a fresh web dictionary at low cost. Hence, the web search queries are widely used in the NLP field. For instance, Hagiwara and Suzuki (2009) used them for a query alteration task, and Sekine and Suzuki (2007) leveraged them for acquiring semantic categories.

More recently, web search clickthrough logs have been explored in the field of lexical acquisition. A web clickthrough is the process of clicking a URL and going to the page it refers. This ensures that the landing page is appropriate since the web user follows the hyperlink after checking the information displayed, such as 'title', 'URL', and 'summary' of their search. Two distinct queries landing on the same 'URL' are possibly input for the same purpose, meaning that they are likely to be related. In the NLP literature, clickthrough logs have been used to learn semantic categories (Komachi et al., 2009) and named entities (Jain and Pennacchiotti, 2010).

The main contribution of this work is two fold:

- We propose a novel method to combine web search query logs and clickthrough logs.
- To the best our knowledge, this work is the first attempt to automatically recognize full spellings given Japanese abbreviations.

This is a very first step of Japanese abbreviation expansion task using search logs.

For evaluation of query expansion method, it is desirable to use a set of queries for evaluation.

[*]The work of Kobayashi were performed at Yahoo Japan. Current affiliation: Google Japan, Roppongi Hills Mori Tower, 6-10-1 Roppongi, Minato-ku, Tokyo 106-6126, Japan

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 410–419,
Chiang Mai, Thailand, November 8 – 13, 2011.

However, it is difficult to obtain them beforehand, because we have to check query logs to find incorrect queries and make necessary changes to define their corrections.

Therefore, in this paper, we focus on query abbreviation and evaluate our proposed approach in an abbreviation expansion task. Abbreviation expansion itself is difficult for many query expansion methods based on edit distance, because the input and output have only a few, if any, characters in common. Our clickthroughlog-based approach can expand even queries that do not share any characters at all with the abbreviated ones [1]. Since our method does not rely on any language, it is applicable to any other languages including Chinese and English.

The rest of this paper is organized as follows. Section 2 describes previous works in query expansion tasks. In Section 3, we formulate a query expansion task in a noisy channel model framework. In Section 4, we show that label propagation on a clickthrough graph can be used as a query abbreviation model and extract candidates for query correction without preparing correct candidates. Section 5 explains the query language model we use. In Section 6, we evaluate our method in an abbreviation expansion task and show its efficiency. Section 7 offers conclusions and directions for future work.

2 Related Work

Query expansion for a web-search query has to handle neologisms and slang on the web. Thus, it is labor-intensive to maintain a list of correctly spelled words for search queries. Additionally, Japanese query expansion includes several tasks, such as word segmentation, word stemming, and acronym expansion. Much of the previous work has focused on each individual task (Ahmad and Kondrak, 2005; Chen et al., 2007; Bergsma and Wang, 2007; Li et al., 2006; Peng et al., 2007; Risvik et al., 2003).

Cucerzan and Brill (2004) clarified problems of spelling correction for search queries, addressing them using a noisy channel model with a language model created from query logs. Gao et al. (2010) and Sun et al. (2010) applied a reranking method applying neural net to the search-query spelling correction candidates obtained from the Cucerzan's method. Their reranking method had the advantageous ability to incorporate clickthrough logs to a translation model learned as a ranking-feature. However, their methods are based on edit distance, and thus they did not deal with the task of synonym replacement and acronym expansion.

Wei et al. (2009) addressed synonym extraction using similarity based on Jensen-Shannon divergence of commonly clicked URL distribution between queries. Their approach is similar to our proposed method, except that they did not use a language model. Also, their method is not scalable and cannot be applied to our task using large-scale data.

Jain and Pennacchiotti (2010) proposed an unsupervised method for named entity extraction from web search query logs. They performed a clustering method using a combination of features based on query logs, web documents, and clickthrough logs. They showed that clickthrough logs give higher accuracy than query logs as a corpus.

Guo et al. (2008) proposed a unified approach for query expansion using a discriminative model. They extended feature function of CRFs (Lafferty et al., 2001) by adding 'operation' to the triplet variables: 'feature', 'label', and 'operation'. 'Operation' represents a process for query expansion. For example, 'operation' can take four states ('deletion', 'insertion', 'substitution', and 'transposition') on spelling correction. However, their method needs supervised data for training and cannot deal with a word that does not occur in the corpus. In fact, they used only 10,000 queries to learn the query expansion model. Unlike their method, our approach takes advantage of an enormous amount of clickthrough logs for learning the query abbreviation model.

Query suggestion is another task that uses search logs (Mei et al., 2008; Cao et al., 2008). Query suggestion differs from our task in that it allows queries to be suggested that are different from the one that the search user types.

Furthermore, some previous works have addressed acquiring a Japanese abbreviation task. Murayama and Okumura (2008) formulated the process of generating Japanese abbreviations by noisy channel model but they did not handle abbreviation expansion. Okazaki et al. (2008) dealt with recognizing Japanese abbreviation tasks as a binary classification problem. They extracted

[1] Note that our method can be applied to query expansion as well.

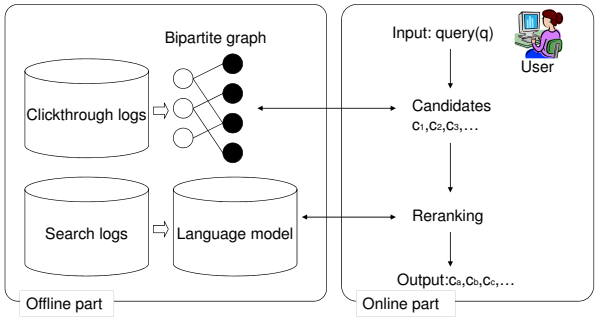

Figure 1: Combining clickthrough logs and search logs for query abbreviation expansion

pairs of words from the newspaper corpus using a heuristic and then classified them as "abbreviation" or "not-abbreviation". However, their heuristic for obtaining abbreviation candidates cannot be applied to web search queries.

3 Noisy Channel Model for Abbreviation Expansion

In this section, we explain our noisy channel based approach to query expansion. We define the query expansion problem as follows: Given a user's query q and a set of search logs L, find a correct query $c \in C$ that is most relevant to the input q. In a probabilistic framework, this can be formulated as finding the $\mathrm{argmax}\, P(c|q)$. Applying Bayes' Rule and dropping the constant denominator, we obtain a unnormalized posterior: $\mathrm{argmax}\, P(c)P(q|c)$(Eq.1). We now have a noisy channel model for query expansion, with two components: the source model $P(c)$ and the channel model $P(q|c)$.

$$\begin{aligned} c^* &= \underset{c}{\mathrm{argmax}}\, P(c|q) \\ &= \underset{c}{\mathrm{argmax}}\, \frac{P(c)P(q|c)}{P(q)} \\ &= \underset{c}{\mathrm{argmax}}\, P(c)P(q|c) \end{aligned} \quad (1)$$

We use a language model estimated from search query logs as the source model, thus $P(c)$ represents likelihood of c as a query. As for the channel model, we use a label propagation method on a clickthrough graph as proposed by Komachi et al. (2009). Figure 1 shows the framework of our approach.

To find candidates to the input query, we construct a bipartite graph from a query and a clicked URL using the web search logs. We calculate the relatedness between the queries on this graph to select a set of candidates C. Since the label propagation is mathematically identical to the random walk with restart, probability of the label propagation can be regarded as the conditional probability $P(q|c)$. If we assume that the relatedness score represents the conditional probability of the typed query q given a candidate $c \in C$, $P(q|c)$, the c^* is calculated by $\mathrm{argmax}\, P(c) \times P(q|c)$. As a consequence, we propose reranking in accordance with the follow equation using two probabilistic models P_{QLM} and P_{LP} and then output ranked candidates. In this paper, we will define P_{LP} interchangeably as a query abbreviation model, P_{QAM}.

$$score(q, c) = P_{QLM}(c) \times P_{QAM}(q|c) \quad (2)$$

An advantage of our proposed method is that it can correct a query by only using search logs without a manually labeled-corpora or any heuristics. Our approach is a versatile framework for query expansion and thus is not specialized for any tasks. We explain the label propagation algorithm on a clickthrough graph and the query language model below.

4 Query Abbreviation Model from Clickthrough Logs

In this section, we describe a label propagation algorithm on a clickthrough graph. It is based on a previous work by Komachi et al. (2009). The main difference between their method and ours is that we use the normalized pointwise mutual information and the 1-step approximation of a clickthrough graph.

Graph-based semi-supervised methods such as label propagation can performance well with only a few seeds and scale up to a large dataset. Figure 2 illustrates the process of label propagation using a seed term "abc".

This is a bipartite graph whose left-hand side nodes are terms and right-hand side nodes are patterns. Starting from "abc", the label propagates to other term nodes through the pattern "http://abcnews.go.com" that is strongly connected to "abc" and thus the label "abc" will be propagated to "american broadcasting corporation".

In this way, label propagation gradually propagates the label of the seed instance to neighboring

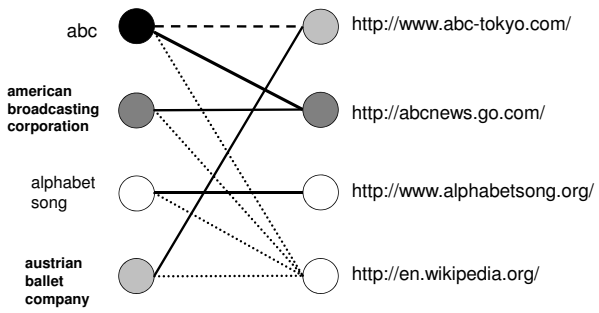

Figure 2: An illustrative example of Instance-Pattern co-occurrence graph and label propagation process.
The strength of lines indicates relatedness between each node, whereas the depth of the color of nodes represents relatedness to the seed. The darker a left-hand side node, the more likely it is similar to "abc". The darker a right-hand side node, the more likely it is the characteristic pattern of "abc".

nodes, and optimal labels are given as the labels at which the label propagation process has converged.

However, the seed instance like that in Figure 2 possibly causes a result to be worse in a task of lexical acquisition, due to an ambiguous instance "abc", which belongs to more than one domain, e.g. "mass media" and "dance". It is expected that the label propagates to unrelated instances if we have highly frequent ambiguous nodes. This problem is called "semantic drift" and has received a lot of attention in NLP research (Komachi et al., 2008).

Komachi et al. (2008) have reported that bootstrapping algorithms like Espresso (Pantel and Pennacchiotti, 2006) can be viewed as Kleinberg's HITS algorithm (Kleinberg, 1999) and the "semantic drift" problem on the graph is the same phenomenon as "topic drift" in HITS, which converges to the eigenvector of the instance-instance similarity graph created from instance-pattern co-occurrence graph as described in the next subsection.

Our label propagation method based on Komachi et al. (2009) can be used as a relatedness measure that returns a similarity score relative to the seed instance, and thus is suitable for a query correction task.

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 3: Laplacian label propagation

4.1 One-step approximation of clickthrough graph

In this paper, we extract queries landing on the same URL as the one related with input query by stopping label propagation after 1-hop. These queries are possibly synonyms with the input query and thus possible to correct without semantic transformation.

Figure 3 shows the label propagation algorithm on a clickthrough graph.

Given an instance set $\mathcal{X} = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ and a label set $\mathcal{L} = \{1, ..., c\}$, the first l instances x_i $(i < l)$ are labeled as $y_i \in \mathcal{L}$. The goal is to predict the labels of the unlabeled instances x_u $(l+1 \leq u \leq n)$.

Let $\mathcal{F}$ denote the set of $n \times c$ matrices with nonnegative entries. A matrix $F = [F_1, \ldots, F_n]^T \in \mathcal{F}$ corresponds to a classification on the dataset $\mathcal{X}$ by labeling each instance x_i as a label $y_i = argmax_{j \leq c} F_{ij}$. Define F_0 as the initial F with $F_{ij} = 1$ if x_i is labeled as a label $y_i = j$ and $F_{ij} = 0$ otherwise. The (i,j)-th element of the final matrix F represents a similarity to the labeled instances. We use these similarities as $P(q|c)$ in Equation 2, where q is a seed instance, c is a labeled instance by label propagation.

The instance-instance similarity matrix A in Figure 3 is defined as $A = W^T W$ where W is an instance-pattern matrix. The (i,j)-th element of W_{ij} contains the relative frequency of occurrence of instance x_i and pattern p_j.

D is a diagonal degree matrix of A where the (i,j)-th element of D is given as $D_{ii} = \sum_j A_{ij}$. Label propagation has a parameter α $(0 \leq \alpha \leq \lambda^{-1}$, where λ is a principal eigenvalue of normalized Laplacian matrix $L)$ that controls the effect of clamping the label distribution of labeled data.

4.2 Normalized PMI

Komachi et al. (2009) suggested that the normalized frequency causes semantic drift (Jurafsky and Martin, 2009), and we confirmed this phenomenon in our preliminary experiment. They suggested using relative frequency such as pointwise mutual information (PMI) and log-likelihood ratio as countermeasure against semantic drift. Therefore, we used pointwise mutual information (PMI) shown below to handle the aforementioned semantic drift problem.

$$PMI(x,p) = ln\frac{P(x,p)}{P(x)P(p)} \tag{3}$$

PMI assigns high scores to low-frequency events. Moreover, using PMI naively makes sparse matrix W dense. Therefore, we used normalized PMI (NPMI) (Bouma, 2009) below as the relative frequency and cut off the values lower than a threshold θ ($\theta \geq 0$).

$$NPMI(x,p) = \left\{ ln\frac{P(x,p)}{P(x)P(p)} \right\} \Big/ -lnP(x,p) \tag{4}$$

$$W_{ij} = \begin{cases} NPMI(x_i,p_j) & (NPMI(x_i,p_j) > \theta) \\ 0 & (NPMI(x_i,p_j) \leq \theta) \end{cases}$$
$$, (\theta \geq 0) \tag{5}$$

NPMI prevents low-frequency events from being assigned scores that are too high by dividing by $-lnP(x,p)$ and heads off excess label propagation through them. By cutting off negative values, the range of W_{ij} can be normalized to [0,1]. Additionally, this prevents sparse matrix W from being dense and reduces the noise in the data.

5 Query Language Model

In this paper, we use a character n-gram language model to obtain the likelihood of the candidates for query expansion in Equation 2.

$$\begin{aligned} P(c) &= \prod_{i=0}^{N-1} P(x_i | x_{i-N+1}, \ldots, x_{i-1}) \\ &= \prod_{i=0}^{N-1} \frac{freq(x_{i-N+1}, \ldots, x_i)}{freq(x_{i-N+1}, \ldots, x_{i-1})} \end{aligned} \tag{6}$$

where consider c is a contiguous sequences of N characters $c = \{x_0, x_1, \ldots, x_{n-1}\}$.

In the web search, neologims appear continuously, which make it hard to compute the likelihood of queries by a word n-gram language model. Moreover, characters themselves carry essential semantic information in Chinese and Japanese. Therefore, we build a character language model for the search query logs following observations of the usefulness of character n-grams for Japanese (Asahara and Matsumoto, 2004) and Chinese (Huang and Zhao, 2006). Asahara and Matsumoto used a window of two characters to the right and to the left of the focus character, which results in using character 5-grams. We also used 5-grams for a query language model from the preliminary experiment.

6 Experiment

6.1 Test Set

We collected abbreviations of *'Acronym'*, *'Kanji'*, *'Kana'* from the Japanese version of Wikipedia, and then removed single letters and duplications. Finally, we gathered 1,916 terms and used them in our evaluation.

6.2 Construction of a Clickthrough Graph

We used queries and clicked links in Japanese clickthrough logs as instances and patterns, respectively. We tallied them in the below conditions.

1. Query and clickthrough are unique with respect to each cookie each day.
   If a user input the same query and clicked the same URL any number of times, we do not count it as occurring multiple times, i.e. we do not increase the number of clickthrough.

2. Alphanumeric characters in a query are unified to one-byte lower-case characters

3. A sequence of white space in a query is unified to single one-byte white space character

4. All the URLs included in clickthrough logs are unique, i.e., we did not generalize URLs as Tseng et al. (2009) did.

The Japanese clickthrough logs were collected from October 22 to November 9, 2009 [2] and from January 1 to 16 in Yahoo Japan web search logs.

[2] A storage device in our experimental environment became full when tallying clickthrough logs. As a result, we were not able to use clickthrough logs of some periods.

Links clicked less than 10 times were removed for efficiency reasons. Finally, we obtained 4,428,430 nodes, 16,841,683 patterns, and 16,988,516 edges.

The threshold θ of elements W_{ij} was set to 0.1 on the basis of preliminary experimental results.

The parameter α for label propagation was set to 0.0001.

6.3 Construction of Query Language Model

We used web search query logs for constructing a language model. The search query logs were collected from August 1, 2009, to January 27, 2010, in Yahoo Japan web search logs. We removed queries that occurred fewer than 10 times. Finally, we obtained 52,399,621 unique queries as a training corpus.

In this experiment, we constructed a character 5-gram language model using the query logs, all normalized by the length of the candidate's string.

6.4 Evaluation

The system output was shown to five search evaluation specialists. We evaluated all systems using *precision* and *coverage at k*. Coverage is defined as the percentage of queries for which the system returned at least one relevant query. Precision at k is the number of relevant queries amongst the top k returned. They are computed as follows:

$$precision = \frac{\text{\# of correct output at rank } k}{\text{Number of output at rank } k},$$
$$coverage = \frac{\text{\# of queries for which the system gives at least one correct output}}{\text{Number of all input queries}} \quad (7)$$

In our experiment, the average number of candidates for each query is about 53. Therefore, we extracted 50 candidates from clickthrough logs and then reranked using three methods:

1. Ranking using abbreviation model (AM) only
2. Ranking using language model (LM) only
3. Ranking using both language model and abbreviation model.

Micro average of edit distance between input abbreviations and its correct expansions is 4.03, while the average length of queries is 3.01. These statistics show that input queries should be replaced by totally different characters and it is difficult to use edit distance for extracting correct candidates from web search logs. This is another reason clickthrough logs are essential to the query abbreviation task.

6.4.1 Judgment Guideline

We describe our guidelines to judge system outputs below. We defined four correction patterns for abbreviation expansion:

1. acronym for its English expansion
2. acronym for its Japanese orthography [4]
3. Japanese abbreviation for its Japanese orthography
4. Japanese abbreviation for its English orthography

We collected abbreviation/expansion pairs if and only if they were one of these three types: (1) named entity, (2) common expression, (3) Japanese meaning of the common expression.

Table 1 shows examples of each correction pattern along with its output type.

Ambiguous cases were discarded in the study as exceptions after discussion with experts. To calculate the agreement rate, system outputs for a hundred randomly sampled queries from test set were evaluated by two judges. The agreement rate of judgment of abbreviation/expansion pair is 47.0 percentage and Cohen's kappa measure $\kappa = 0.63$. Thus, it is considered as an upper bound of the system, and the abbreviation expansion is not considered to be a trivial task.

6.5 Experimental Results

Table 2 shows *precision at k* and coverage for three systems with *k* ranging from 1 to 50. Table 3 shows examples of inputs and outputs. The baseline without reranking is shown at the bottom line (k=50). The result of using only QAM in Table 2 is equivalent to the method of Komachi et al. (2009) using NPMI instead of raw frequency as elements of an instance-pattern matrix. To

[3] Underlined words are correct.

[4] Some corrections were dealt with as exceptions. For example, acronym for its Japanese *Hiragana* was treated as incorrect, but acronym for its Japanese meaning was treated as correct.

Table 1: Abbreviations and its correction

abbreviation	correct candidates（descending order of rank）	output type	correction pattern
adf	asian dub foundation	Named Entity: Organization	acronym for its expansion
ana	全日空，全日本空輸株式会社 (All Nippon Airways)	Named Entity: Organization	acronym for its Japanese orthography
ny	ニューヨーク (New York)	Named Entity: Location	acronym for its Japanese orthography
tos	テイルズオブシンフォニア (Tales of Symphonia)	Named Entity: Product	acronym for its Japanese orthography
イラレ	illustrator	Named Entity: Product	Japanese abbreviation for its English orthography
ハンスト	ハンガーストライキ (Hunger Strike)	Common expression	Japanese abbreviation for its Japanese orthograpy
阪神	阪神タイガース	Named Entity: Organization	Japanese abbreviation for its Japanese orthograpy
fyi	for your information	Common expression	acronym to its expansion

Table 2: Precision and coverage at *k*

k	query abbreviation model (QAM)		query language model (QLM)		QLM+QAM	
	precision	coverage	precision	coverage	precision	coverage
1	0.114	0.114	0.157	0.157	**0.161**	**0.161**
3	0.122	0.256	0.142	0.278	**0.157**	**0.321**
5	0.121	0.341	0.128	0.346	**0.142**	**0.392**
10	0.114	0.453	0.102	0.425	**0.115**	**0.465**
30	**0.087**	0.536	0.078	0.529	0.082	**0.542**
50	0.073	0.557	0.073	0.557	0.073	0.557

Table 3: Examples of input and candidates or its correction[3]

Input	Candidates
写植	写真植字，写植屋，写植機，写植 方，漫画
満鉄	満鉄調査部，南満州鉄道株式会，南満州鉄道，満鉄会，満州鉄道
はねトび	はねるのとびら，はねるのトびら，はねるの，はねるのトびら，はねるのトビラ，はねとび
vod	ビデオオンデ，ビデオ・オン・デマンド，ビデオ オンデマンド，ビデオオンデマンド
ilo	日本 ilo，ilo 協会，国際労働機関，国際労働期間，ilo 条約
pr	パブリック・リレーションズ，パブリックリレーションズ，prohoo!マ，pr 会社，プラ

Table 4: P-values of Wilcoxon's signed rank test

	QAM and QAM + QLM	QLM and QAM + QLM
p-value	0.055	$7.79e^{-10}$

our knowledge, their algorithm is the state-of-the-art algorithm in acquiring synonyms using web search logs.

The proposed ranking method using a query language model and abbreviation model learned from clickthrough logs shows the best precision and coverage within $1 \leq k \leq 10$. This is because the language and abbreviation model use different sources of information to complement each other.

The language model estimates probability of the candidate as a query, and it assigns high probability to candidates that appear frequently in query logs. Those candidates tend to co-occur with many clickthrough patterns, which results in creating generic patterns that may cause semantic drift (Komachi et al., 2009). Because we used NPMI instead of raw frequency, our label propagation method assigns high weight to instances connected to a seed instance through a few specific patterns. Consequently, low-frequency instances tend to be ranked in higher positions.

Table 4 shows the significance level between two baselines and the proposed model. We applied Wilcoxon's signed rank test to compare harmonic mean between precision and coverage of each model with *k* ranging from 1 to 50. The improvement of adding QAM to QLM is made statistically significant by the Wilcoxon's signed rank test at level $p < 0.00001$. Our approach outperforms the QAM without QLM although not as significant ($p < 0.06$). These mean that the ranking of our methods is similar to that of QAM. We consider the reason or this result to be that QLM introduces more information about queries under this experimental setting because the reranking process is performed after narrowing candidates down

to 50 by QAM, even though we do not use QAM scores at all when evaluating QLM.

Due to time constraints and human resources for evaluation, we were unable to compare NPMI with raw frequency. There is still much room for improvement for assigning appropriate weights to edges in a clickthrough graph.

6.6 Error Analysis

We conducted error analysis of our proposed method and found that errors can be divided into three types: (1) a partial correct query, (2) a correct query but with an additional attribute word, and (3) a related but not abbreviated term.

A partial correct query The main reason for this error is that the likelihood of the partial query becomes higher than that of its correct spelling. Although we normalized the likelihood of candidates by their string length, we still fail to filter fragments of queries. We consider that this issue can be solved by modeling popularity of candidates using PageRank from web search logs. Partial correct queries do not co-occur with attribute words frequently, while correct queries co-occur with diverse attribute words. Therefore, PageRank on a query graph whose edges represent common co-occurring words between queries, will assign higher scores to correct queries than a query language model and abbreviation model.

A correct query but with an additional attribute word Examples of this error type include the combination of correct queries and commonly used attribute words in the search (e.g.“* とは”*(what does * mean?)*, “* 意味”*(* meaning)*,“* 使い方” *(how to use *)*, etc.). There were 857 queries that were classified as incorrect that co-occurred with these attribute words. The similarity of these candidates and input query tend to be higher than that of others because these attribute words frequently appear in search query logs, so the likelihood of these candidate being calculated by a language model tends to be higher too.

We consider that this issue can be solved at some level by generating a language model using the first term only, after splitting queries separated by a space in search query logs. However, attribute words are not always separated by a space, and sometimes appear as the first term in the query[5]. Another way to handle this problem is to use PageRank described earlier to decrease likelihood of candidates including attribute words.

Related but not abbreviated term A number of abbreviations coincide with other general nouns (e.g. “dog”[6]). It is hard to expand these abbreviations correctly at present. In future work, session logs and geo-location information from IP address and GPS can be used to disambiguate the intent of the query.

Besides above reasons, 280 out of 1,916 queries did not exist in clickthrough logs, resulting in our system not being able to extract the correct query. To solve this problem, we will increase clickthrough logs to improve the coverage of our corpus.

7 Conclusion

We have proposed a query expansion method using the web search query and clickthrough logs.

Our noisy channel based method uses character 5-gram of query logs as a language model and label propagation on a clickthrough graph as a channel model. In our experiment, we found that a combination of label propagation and language model outperformed other methods using either label propagation or language model in reranking of query abbreviation candidates extracted from the web search clickthrough logs.

In fact, a modified implementation of this method is currently in production use as an assistance tool for making a synonym dictionary at Yahoo Japan.

In evaluation of IR systems, Mizzaro (2008) has proposed a normalized mean average precision (NMAP) for considering difficulty of topics in data sets. However, identifying topics in our test set queries and measuring their difficulty are beyond the scope of this paper. Evaluation criteria are important for making production services.

In the future, we are going to address this task using discriminative learning as a ranking problem.

References

Farooq Ahmad and Grzegorz Kondrak. 2005. Learning a spelling error model from search query logs. In

[5]Some attributes, (e.g. “動画”(*Movie*),“’ アニメ’(*Animation*),“画像”(*Picture*),etc.), occur often at first token in a search query, but some attributes, (e.g. “使い方”, ,“意味”,etc.) almost never occur at first token.

[6]DOG: Disk Original Group

Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 955–962.

Masayuki Asahara and Yuji Matsumoto. 2004. Japanese unknown word identification by character-based chunking. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 459–465.

Shane Bergsma and Qin Iris Wang. 2007. Learning noun phrase query segmentation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Geolof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, pages 31–40.

Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 875–883.

Qing Chen, Mu Li, and Ming Zhou. 2007. Improving query spelling correction using web search results. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 181–189.

Silviu Cucerzan and Eric Brill. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 293–300.

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 358–366.

Jianfeng Guo, Gu Xu, Hang Li, and Xueqi Cheng. 2008. A unified and discriminative model for query refinement. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 379–386.

Masato Hagiwara and Hisami Suzuki. 2009. Japanese query alteration based on seamntic similarity. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 191–199.

Chang-Ning Huang and Hai Zhao. 2006. Which is essential for chinese word segmentation: character versus word. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC-20)*, pages 1–12.

Alpa Jain and Marco Pennacchiotti. 2010. Open entity extraction from web search query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 510–518.

Daniel Jurafsky and James H. Martin. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. 2nd edition. Prentice Hall, Englewood Cliffs. NJ.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, September.

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1020.

Mamoru Komachi, Shimpei Makimoto, Kei Uchiumi, and Manabu Sassano. 2009. Learning semantic categories from clickthrough logs. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 189–192.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Mu Li, Yang Zhang, Muhua Zhu, and Ming Zhou. 2006. Exploring distributional similarity based models for query spelling correction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1025–1032.

Qiazhu Mei, Dengyong Zhou, and Kenneth Church. 2008. Query suggestion using hitting time. In *Proceeding of the 17th ACM conference on Information and Knowledge Management*, pages 469–478.

Stefano Mizzaro. 2008. The good, the bad, the difficult, and the easy: something wrong with information retrieval evaluation? In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 642–646, Berlin, Heidelberg. Springer-Verlag.

Norihumi Murayama and Manabu Okumura. 2008. Statistical model for Japanese abbreviations. In *Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, pages 260–272.

Naoki Okazaki, Mitsuru Ishizuka, and Jun'ichi Tsujii. 2008. A discriminative approach to Japanese abbreviation extraction. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP-08)*, pages 889–894.

Patric Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 113–120.

Fuchun Peng, Nawaaz Ahmed, Xin Li, and Yumao Lu. 2007. Context sensitive stemming for web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 639–646.

Knut M. Risvik, Tomasz Mikolajewski, and Peter Boros. 2003. Query segmentation for web search. In *Poster Session in The Twelfth International World Wide Web Conference*.

Satoshi Sekine and Hisami Suzuki. 2007. Acquiring ontological knowledge from query logs. In *Proceedings of the 16th international conference on World Wide Web*, pages 1223–1224.

Xu Sun, Jianfeng Gao, Daniel Micol, and Chris Quirk. 2010. Learning phrase-based spelling error models from clickthrough data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 266–274.

Huihsin Tseng, Longbin Chen, Fan Li, Ziming Zhuang, Lei Duan, and Belle Tseng. 2009. Mining search engine clickthrough log for matching N-gram features. In *Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing*, pages 524–533.

Xing Wei, Fuchunk Peng, Huishin Tseng, Yumao Lu, and Benoit Dumoulin. 2009. Context sensitive synonym discovery for web search queries. In *Proceeding of the 18th ACM conference on Information and Knowledge Management*, pages 1585–1588.