SNS Retrieval Based on User Profile Estimation Using Transfer Learning from Web Search

Daisuke Kataoka
Graduate School of Informatics
Kyoto University
Kyoto, Japan
kataoka@dl.soc.i.kyoto-u.ac.jp

Keishi Tajima
Graduate School of Informatics
Kyoto University
Kyoto, Japan
tajima@i.kyoto-u.ac.jp

***Abstract*—In this paper, we propose a method of retrieving posts on social networking services (SNSs) by specifying a pair of queries: a topic query and an entity query. A topic query specifies the topic of the posts to retrieve (e.g., "iPhone") and an entity query specifies the type of users who posted them (e.g., "students"). In the existing search systems for SNS posts, we can specify topics of posts by keywords, but we cannot specify types of users. Even if we include keywords specifying types of users in a query, such keywords are not usually included in tweets or user profile data. In our method, we estimate types of users by learning vocabulary whose appearance is correlated with specific types of users. We learn it from the datasets obtained through Web search. We retrieve Web documents through the search with a keyword specifying the type of users (e.g., "student"), and we also retrieve Web documents by using a keyword specifying its opposite (e.g., "adult"). We regard the documents retrieved by these queries as positive and negative examples of documents describing the target type, and we train a model for recognizing users of the given type. We recognize users of the target type by inputting their posts and their profile data into the model. We use Web documents instead of SNS posts for training the model because the Web has more documents describing types of people.**

***Index Terms*—microblog, profile estimation, transfer learning**

1. Introduction

The wide-spread use of social network services (SNSs) enables us to broadcast and share information about our opinions, activities, and statuses. *Twitter* is one of the most popular SNSs, which is regarded as a microblogging service owing to its length limitation of posts. In Twitter, each post, which is called a tweet, may contain at most 140 characters. This characteristic has encouraged users to post their opinions frequently and immediately, and has succeeded in eliciting a wide range of information from users. Companies have been interested in mining opinions from huge data of SNSs, in particular, honest opinions about their own events or their new products.

One of the most important challenges in mining opinions from SNSs is the incompleteness of the user profile information, which has caused difficulty in identifying users' profile (e.g., age or sex) for search systems. A straightforward way for a company that wants to know opinions of students on a new product "X" would be to search Twitter for tweets related to "X" posted by students. A simple keyword query "X AND students', however, would result in a low quality result both in precision and recall because tweets including the keyword "student" are not necessarily posted by students, and tweets posted by students do not usually include the keyword "students". Even profile data of student users usually do not include the keyword "students". If a user simply submit a query "X", the user have to find tweets from students by manually scrutinizing their profiles or past tweets. This example indicates that the incompleteness of user profile information in SNS can lead to low recall and precision, in particular, while targeting users with a particular type of profile.

In this study, to address this challenge in SNS opinion mining, we propose a search method that integrates user profile estimation. Our method enables us to find posts containing a topic (e.g. "iPhone") and posted by a specific type of users (e.g., "students"). In our method, a user submits a pair of queries: a *topic query* and a *profile query*. A topic query specifies the topic of the posts to retrieve (e.g., "iPhone") and an entity query specifies the type of users who posted them (e.g., "student"). In order to estimate types of users, we train a classifier whose inputs are their posts and their profile data. To train the classifier, we use data obtained through Web search. We retrieve Web documents with queries consisting of keywords given in the entity query (e.g., "student"), and we also retrieve Web documents by using a keyword specifying the opposite type of people (e.g., "adult"). We then regard the documents retrieved by these queries as positive and negative examples of documents describing the target type, and we train the classifier by using these data.

There are two reasons why we use Web documents instead of SNS posts for the training. First, Web pages are usually longer than SNS posts, and we expect that Web pages include more useful vocabulary than SNS posts. Secondly, the Web has more documents describing types of people, such as students. For example, if we retrieve

Web documents by using a keyword "student", the result would include more documents describing "students", such as Wikipedia[1] articles, and such pages must contain vocabulary such as "education" or "campus". On the other hand, Twitter posts retrieved by using a keyword "student" would have less useful vocabulary. Even if we retrieve user profile data including the keyword "student", we cannot obtain vocabulary whose appearance is strongly correlated with "students" because a user profile data is usually a set of keywords which independently describe several different aspects of the user.

To demonstrate the effectiveness of our proposed method, we build a test dataset comprising three pairs of entity queries (each pair consists of two entity queries specifying the opposite types of users) and five topic queries for each of the three entity query pairs. In total, we have 3*2*5=30 combinations of a entity query and a topic query. We retrieved the top 100 tweets for each topic queries by using the search function of Twitter, and labeled those tweets by manually examining if they match with the entity queries.

The results of experiments on this dataset showed that our proposed method outperforms the baseline method in the most cases. We also compared the performance of our method with changing the number of documents used in the training, and the result shows that the optimal number of documents for the training is different for each query.

The primary contributions of this research is summarized as follows:

- First, we introduce the problem of retrieving SNS posts that are posted by a particular type of users.
- Second, we propose an effective method for user profile estimation based on a classifier trained by the dataset obtained through Web search, which make it easy to collect datasets for training a classifier.
- Third, we constructed a test dataset and demonstrated the effectiveness of our proposed methods. In particular, we showed that training data obtained through Web search is superior than that obtained through Twitter search. We also showed that the optimal number of documents used for the training is different for each query.

The remainder of this paper is organized as follows: Section 2 surveys related work; Section 3 describes our proposed method in detail. Section 4 demonstrates the effectiveness of our method on our test dataset. Finally, Section 5 presents a summary of this work and conclusions.

2. Related Work

Several SNS research avenues are related to our proposed method. In microblogs, it is often unclear what a user is talking about in a short post, and it is one of the main issues in microblog retrieval. Several studies have proposed methods of improving the performance of microblog retrieval by using features unique to SNSs, e.g, followers, replies, hashtags [1], [2], [3], [4], [5]. Luo et al. [2] showed that information specific to Twitter, such as hashtags and mentions, can improve Twitter retrieval performance, while Nagmoti et al. [1] used the number of followers and following-follower ratio as an indicator of the authority degree of users, and used it for improving the performance of tweet retrieval.

There have also been many studies on methods for estimating some attributes of users or posts [6], [7]. Pochampally and Varma [6] proposed a method that determines topics of interest of given Twitter users based on their context information. Lu et al. [8] proposed a method of identifying entities indirectly mentioned in SNS posts. Their method uses information on other mentions in the posts, and also uses vocabulary taken from Wikipedia articles describing the entities in order to compensate for insufficient information in short posts. Our method also often uses vocabulary taken from Wikipedia pages because we use vocabulary in Web pages included in the results of Web search with a keyword in our entity query, and the results of Web search with such a keyword often rank some Wikipedia pages high.

In our method, profiles of Twitter users are estimated by a classifier trained on a dataset of Web domain. Several studies have tried such knowledge propagation across different domains [9], [10]. In machine learning, such propagation is called *transfer learning* [11]. Transfer learning focuses on storing knowledge obtained while solving one problem and applying it to a different but related problem. Peddinti et al. [12] performed sentiment analysis of Twitter posts about movies by transferring knowledge from movie review domain. Rieman et al. [13] proposed a domain adaptation method that adjusts the observed word counts in the target domain, leaving the source domain model unchanged, for the problem of applying Facebook user-level language models to country-level Twitter language.

Relevance feedback [14] or pseudo-relevance feedback (PRF) is a representative method of query expansion, whose effectiveness has shown also on microblog retrieval [15], [16], [17], [18], [19]. PRF is a derivative of relevance feedback, where top-ranked documents are regarded as a positive examples of the query. Our proposed method can be regarded as a variation of PRF because we regard documents in the results of Web search as positive examples and extract vocabulary from them. Whiting et al. [20] demonstrated that PRF can improve retrieval performance. In another paper [16], Whiting et al. also showed that performance of microblog retrieval can further be improved by combining PRF with a word-weighting scheme that uses PageRank and temporal properties of words. Choi and Croft [17] proposed a method for selecting specific time periods for selecting tweets for PRF in tweet retrieval.

The problem we address in this paper is similar to that of Kataoka et al. [15]. They also addressed a problem of retrieving microblog posts by specifying a topic and a user type. In the method proposed in [15], however, the searcher needs to provide feedbacks on the search result obtained by a topic query. On the other hand, our proposed method employs a classifier trained on the datasets obtained through

1. https://www.wikipedia.org/

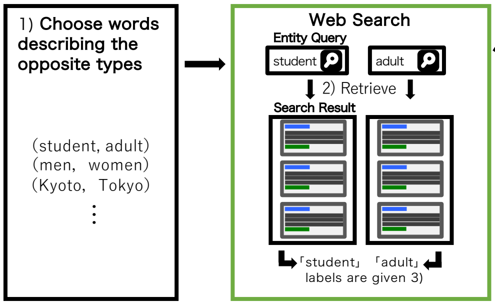

Figure 1. Training Phase

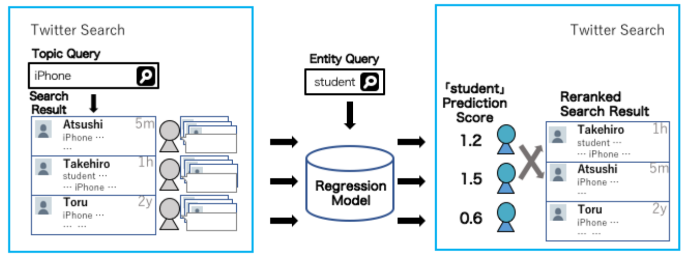

Figure 2. Application Phase

Web search.

3. Proposed Method

3.1. Problem settings

In this section, we explain the problem settings in our research. In our method, a user specifies a pair of queries: a topic query and an entity query. A topic query specifies the topic of the posts to retrieve (e.g., "iPhone"), while an entity query specifies the type of users who posted them (e.g., "student"). Our purpose is to retrieve documents that include contents matching with the topic query and are posted by users matching with the entity query.

The problem in retrieving SNS posts by both a topic query "iPhone" and an entity query "student" is that posts written by students rarely include the keyword "student". However, the vocabulary related to students (e.g., "examination" and "class") often appears in the context of the posts, i.e., in the past tweets and the profile data of the user. Our idea is that types of users can be estimated by a classifier whose input is features extracted from the contents of posts and their context data.

In order to train such a classifier, we use a dataset obtained through *Web search* instead of *Twitter search*. It is because we expect that Web pages obtained through Web search include more vocabulary for identifying student users than tweets obtained through Twitter search. There are two reasons why we expect it. First, most Web pages are longer than tweets and include more vocabulary. Second, Web pages more often include sentences describing students than tweets because tweets more often include conversations, and sentences describing students must include more useful vocabulary for our classifiers.

Our method consists of two phases: a training phase and an application phase. We will explain the details of these two phases in this section.

3.2. Training Phase

In the first phase, we train a classifier on a dataset obtained through Web search. This phase consists of the following steps.

1) Given a keyword describing a type of users, we choose a keyword describing the opposite type of users. Such a keyword is ideally an antonym of the given keyword, but it can also be some co-hyponym when we do not have an antonym. For example, when we are given a keyword "student", "adult" is a good candidate.
2) Submit a query consisting of the given keyword and the paired query with the negation operator to a Web search engine. For example, when given a keyword "student", we submit a query "student -adult". Also submit a query with the keyword and the paired keyword exchanged, that is, "adult -student".
3) We retrieve top-n documents in the answers of these queries, and use them as the positive and negative examples, respectively, for the training of the classifier.
4) Train a classifier by using the datasets obtained through the two queries above.

Figure 1 illustrates these steps in the training phase.

In this paper, we assume the user specifies a keyword specifying a type of user and a keyword specifying the opposite type, but we can extend it to three or more types. When retrieving Web pages, we combine a keyword and its opposite with the negation operator, e.g., "student -adult". It is because the result of a simple query "student" may include many pages related to both students and adult people, e.g., pages comparing them, and such pages would become noise data in the training. In order to eliminate such pages from the dataset, we use the query "student -adult".

3.3. Application Phase

Next, we explain the second phase of our method: the application phase. In the application phase, we retrieve candidate posts and their context data, input them to the classifier trained in the previous phase, and rank the candidate posts based on their likelihood of being posted by the target type of users. In this paper, we apply our method to tweets retrieval from Twitter. This phase consists of the following steps:

1) Takes a topic query and an entity query from the user.

2) Submit the topic query to a Twitter search engine, and retrieve tweets containing the keyword in the topic query as the candidate tweets.
3) Calculate a document vector for each candidate tweet.
4) Input the vectors into the classifier and obtain prediction scores.
5) Rank the candidate tweets by their prediction scores.

An example of this process is shown in Figure 2. In this example, a topic query is “iPhone” and an entity query is “student”. A document vector for each tweet in the search result is calculated by the method proposed in Kataoka et al. [15]. The method calculates the vector based on the contents of the tweet, and also on its context information, which includes the profile of the user who posted it, the past tweets of the user, and the profile of its followers. We omit the details of the method, but it calculates the vector by a weighted sum of the TF-IDF vectors calculated from the contents of the tweet and the context information.

The TF-IDF value for a tweet d and a term t is calculated by the following formula:

$$w(d,t) = \mathrm{tf}_{d,t} \cdot (\mathrm{idf}_t + 1)$$

where $\mathrm{tf}_{d,t}$ is the number of occurrences of word t (the term frequency), and idf_t is defined by the following formula:

$$\mathrm{idf}_t = \log_2 \frac{|D|}{n_t}$$

where D is the set of all documents and n_t is the number of documents that include word t.

4. Experiments

We evaluate our method through the following two experiments.

- We compare the performance of the method that uses Web pages for the training and that of the method that uses tweets.
- We compare both methods with different size of training dataset.

When we retrieve data for training through Web search or Twitter search, we retrieve top-n answers. By changing the value of n, we obtain different size of training datasets. When n is large, we can use a larger training dataset, which usually improves the quality of the trained model. On the other hand, when n is large, the quality of top-n results could be lower. For example, if we use top 10 results of a Web search, we can expect that all the 10 web pages are relevant to the query keyword, but if we use top 640 results, it may include many irrelevant pages. Optimal size of the training dataset must be determined by the trade-off between these two factors.

We also examine what kind of words are identified as effective for each entity query in each method.

4.1. Datasets and Evaluation measure

We chose the following three pairs of entity queries: (student, adult), (man, woman), (Tokyo, Osaka). They correspond to three major demographic attributes of people: age, sex, and region. For each pair of entity queries, we chose five topic queries, which results in 3*2*5=30 combinations of an entity query and a topic query. Some examples of topic and entity queries are shown in the Table 1. For example, the user intentions of q_5 and q_6 in Table 1 is collecting information on reputation of Star Wars among male users and female users, respectively.

For each topic query, we retrieved top 100 tweets by submitting a keyword in the topic query to Twitter REST API[2], and labeled them by manually examining whether they match with the each of the entity queries. For example, we retrieve 100 tweets by a keyword query “smart speaker”, and classified them into “student”, “adult”, and “unclear”.

When we manually label a tweet for entity queries, we read the contents of the tweet, the profile of the user who posted it, and also the profile of the followers. We examine if there are enough clues to determine that the user matches with the entity queries, and if there are not enough clues, we assign the label “unclear”. For example, if the user is talking about “campus” and “dormitory” in the past tweet or if the proportion of the followers whose profile include the word “university” is large, they are clues for determining the user is a student. The column P@100 in Table 1 shows the number of tweets that were determined to be relevant among the 100 tweets.

When we collect tweets for training datasets, we also used Twitter REST API[3], and when we collect web pages for training datasets, we used Bing Web Search API[4].

When we create a document vector of a candidate tweet, we include the profile of the user who posted it, the profiles of its followers (at most 1200 followers), and its posts before and after the candidate tweet (at most 25 tweets before the tweet and at most 25 tweets after the tweet). When we create a feature vector of a tweet in the training datasets, we also include its contexts, while when we create a feature vector of a web page in the training datasets, we only use its contents. Table 2 shows a statistics on the vocabulary in the training datasets obtained from Twitter and the Web.

Random Forests [21] are employed for learning a classifier, which are a combination of decision trees such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.

To evaluate retrieval effectiveness of methods, the *normalized Discounted Cumulative Gain* (nDCG) was used. The nDCG is an evaluation index based on multivalued relevance, which allowed the performance measurements to take account of graded relevance. The nDCG@k only looks

2. https://dev.twitter.com/rest/public
3. https://dev.twitter.com/rest/public
4. https://azure.microsoft.com/ja-jp/services/cognitive-services/bing-web-search-api

TABLE 1. EXAMPLES OF RETRIEVAL TASKS

	Topic Query	Entity Query	Entity	P@100
q_1	Smart speaker	student	age	2
q_2	Smart speaker	adult	age	52
q_3	Monster Hunter	student	age	42
q_4	Monster Hunter	adult	age	58
q_5	Star Wars	man	sex	59
q_6	Star Wars	woman	sex	35
q_7	influenza	man	sex	34
q_8	influenza	woman	sex	46
q_9	Asakusa	Tokyo	region	31
q_{10}	Houses of Parliament	Tokyo	region	31
q_{11}	Koshi-En	Osaka	region	16
q_{12}	Osaka-Jo Park	Osaka	region	42

TABLE 2. COMPARISON OF VOCABULARY

	Average	Max	Min
Twitter	2177.6	4607	62
Web	1147.9	14215	1

at the top k ranks and is an index obtained by normalizing DCG@k [22]; DCG@k is defined as follows:

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{1}$$

Here, rel_i is the degree of relevance of the i-th search result. In this experiment, we set $rel_i = 1$ if the i-th document was relevant, and $rel_i = 0$ if it was irrelevant.

4.2. Experiment Details

In our experiment, we compare the following four methods.

1) **Twitter**: Training datasets are obtained through Twitter retrieval
2) **Twitter Neg**: We add the negation of the opposite type keyword to a query of the **Twitter**
3) **Web**: Training datasets are obtained through Web retrieval
4) **Web Neg**: We add the negation of the opposite type keyword to a query of the **Web**

In the **Twitter Neg** method and the **Web Neg** method, the query for collecting datasets through Twitter or Web retrieval consists of two terms, one is the keyword in the entity query and the other is the keyword representing the opposite type with the negation operator. For example, when we create a dataset for an entity query “student” , we use a query “student -adult” in **Twitter Neg** and **Web Neg**.

Furthermore, we change the value of n, i.e., the size of the training datasets, to $10 * 2^i$ for $0 \leq i \leq 6$. As explained before, we expect that the increase of n does not necessarily lead to acquisition of a better training dataset. Therefore, by changing n, we discover the optimal value of n with the best search performance.

The experimental results of our method are shown in Table 3 to Table 14. The k in the first column of the tables represent k in nDCG@k, and the numbers 10 to 640 in the top row represent n. Each table represents the average of the results of applying each method to 10 queries (combinations of 5 topic queries and a pair of entity queries corresponding to either age, sex, or region). Table 3 to Table 6 show the results of the four methods for the entity query pair for “age”, Table 7 to Table 10 show the results of the four methods for the entity query pair for “sex”, and Table 12 to Table 14 show the results of the four methods for the entity query pair for “region”.

We also show these results in the graphs shown in Figure 3 to Figure 8. The vertical axes of these graphs represent nDCG@k, and the horizontal axes represent n. These graphs show that **Web Neg** outperforms other methods on nDCG@10 and nDCG@30 of the entity query category “age”, and on nDCG@10 of the entity query category “region”, while **Twitter** outperforms other methods on nDCG@10 and nDCG@30 of the entity query category “sex”. **Web** outperforms other methods on nDCG@30 of the entity query category “region”.

4.3. Discussions

This section discusses the results shown above. Our experiments revealed that the performance of our method differs for each n and each entity query. However, although the training datasets obtained through Twitter search have more vocabulary than the datasets obtained through the Web search as shown in Table 2, the classifiers trained by the latter outperformed the classifiers trained by the former for the entity query categories “age” and “region”.

We examined what words are regarded as effective features in the Random Forest models by comparing the importance of the features by the method described in [23]. The importance is calculated through ”gini importance”, which is defined by the total decrease in node impurity (weighted by the probability of reaching that node, which is approximated by the proportion of samples reaching that node) averaged over all trees of the ensemble.

As an example, we compare the feature importances of the classifier for “age” categories trained by the four datasets obtained by the four methods, **Web** and **Web Neg**, **Twitter**, and **Twitter Neg**, respectively. In this comparison, we fix the value of n to 320. The column “Rank” represents the ranks of the feature importances and the columns “score” represent the “gini importance” of each feature. Compared with **Twitter** and **Twitter Neg**, **Web** and **Web Neg** could find better features that are obviously relevant to “age” properties of people.

In future work, we plan to automatically tune the value of n for each entity query because the optimal n is significantly different for each query and each method. Furthermore, the Web domain has a distribution of vocabulary different from that in the Twitter domain. We are considering

TABLE 3. Web Age Entity Comparison

k	10	20	40	80	**160**	**320**	640
3	0.300	0.400	0.347	0.306	**0.459**	**0.530**	0.300
5	0.301	0.372	0.334	0.319	**0.487**	**0.492**	0.326
10	0.283	0.397	0.39	0.342	**0.446**	**0.475**	0.347
20	0.318	0.395	0.383	0.347	**0.420**	**0.448**	0.369
30	0.333	0.386	0.403	0.376	**0.416**	**0.444**	0.386

TABLE 4. Web Neg Age Entity Comparison

k	10	20	**40**	80	160	**320**	640
3	0.230	0.230	**0.535**	0.353	0.447	**0.589**	0.577
5	0.318	0.265	**0.455**	0.340	0.450	**0.579**	0.526
10	0.333	0.314	**0.415**	0.319	0.425	**0.522**	0.467
20	0.340	0.306	**0.410**	0.323	0.420	**0.490**	0.423
30	0.340	0.306	**0.397**	0.346	0.429	**0.484**	0.453

TABLE 5. Twitter Age Entity Comparison

k	10	20	40	80	**160**	**320**	**640**
3	0.265	0.283	0.323	0.277	**0.323**	**0.323**	**0.323**
5	0.303	0.314	0.302	0.322	**0.358**	**0.358**	**0.358**
10	0.318	0.35	0.313	0.306	**0.365**	**0.365**	**0.365**
20	0.35	0.359	0.322	0.342	**0.360**	**0.360**	**0.360**
30	0.37	0.434	0.331	0.352	**0.373**	**0.373**	**0.373**

TABLE 6. Twitter Neg Age Entity Comparison

k	10	20	40	80	**160**	**320**	**640**
3	0.23	0.27	0.323	0.283	**0.323**	**0.323**	**0.323**
5	0.22	0.251	0.343	0.301	**0.358**	**0.358**	**0.358**
10	0.24	0.318	0.347	0.328	**0.365**	**0.365**	**0.365**
20	0.318	0.35	0.368	0.332	**0.360**	**0.360**	**0.360**
30	0.354	0.375	0.371	0.351	**0.373**	**0.373**	**0.373**

TABLE 7. Web Sex Entity Comparison

k	**10**	20	**40**	80	160	320	640
3	**0.517**	0.283	**0.536**	0.311	0.323	0.300	0.477
5	**0.470**	0.287	**0.497**	0.364	0.358	0.354	0.522
10	**0.402**	0.351	**0.495**	0.363	0.359	0.401	0.457
20	**0.401**	0.363	**0.478**	0.405	0.330	0.422	0.453
30	**0.421**	0.380	**0.516**	0.413	0.337	0.434	0.475

TABLE 8. Web Neg Sex Entity Comparison

k	10	20	40	80	**160**	320	640
3	**0.436**	0.347	0.377	0.388	**0.530**	0.359	0.230
5	**0.414**	0.346	0.355	0.419	**0.466**	0.385	0.306
10	**0.457**	0.352	0.364	0.412	**0.442**	0.412	0.320
20	**0.460**	0.347	0.375	0.400	**0.421**	0.432	0.391
30	**0.461**	0.370	0.396	0.411	**0.404**	0.439	0.410

TABLE 9. Twitter Sex Entity Comparison

k	**10**	20	40	80	160	320	**640**
3	**0.547**	0.500	0.335	0.464	0.423	0.489	**0.530**
5	**0.546**	0.430	0.380	0.435	0.402	0.489	**0.508**
10	**0.546**	0.427	0.418	0.415	0.435	0.450	**0.471**
20	**0.522**	0.429	0.401	0.400	0.410	0.422	**0.450**
30	**0.526**	0.432	0.415	0.418	0.425	0.431	**0.482**

TABLE 10. Twitter Neg Sex Entity Comparison

k	**10**	20	40	80	160	**320**	640
3	**0.506**	0.347	0.417	0.406	0.406	**0.528**	0.377
5	**0.503**	0.372	0.426	0.403	0.39	**0.509**	0.383
10	**0.510**	0.383	0.421	0.395	0.427	**0.443**	0.432
20	**0.520**	0.414	0.399	0.386	0.404	**0.406**	0.429
30	**0.509**	0.433	0.414	0.419	0.421	**0.408**	0.457

TABLE 11. Web Neg Region Entity Comparison

k	**10**	20	**40**	80	160	320	640
3	**0.300**	0.217	**0.470**	0.236	0.123	0.212	0.159
5	**0.312**	0.186	**0.382**	0.240	0.183	0.194	0.156
10	**0.279**	0.238	**0.338**	0.232	0.178	0.252	0.207
20	**0.275**	0.243	**0.307**	0.237	0.205	0.260	0.247
30	**0.292**	0.294	**0.336**	0.308	0.266	0.259	0.248

TABLE 12. Web Region Entity Comparison

k	**10**	**20**	40	80	160	320	640
3	**0.347**	**0.353**	0.283	0.200	0.170	0.259	0.212
5	**0.330**	**0.298**	0.286	0.242	0.179	0.257	0.209
10	**0.323**	**0.272**	0.322	0.267	0.218	0.272	0.246
20	**0.351**	**0.286**	0.299	0.232	0.211	0.249	0.227
30	**0.368**	**0.329**	0.324	0.265	0.229	0.259	0.264

TABLE 13. Twitter Region Entity Comparison

k	**10**	**20**	40	80	160	320	640
3	**0.253**	**0.247**	0.200	0.141	0.153	0.153	0.130
5	**0.238**	**0.247**	0.215	0.131	0.195	0.195	0.165
10	**0.238**	**0.243**	0.206	0.154	0.219	0.219	0.181
20	**0.247**	**0.248**	0.210	0.196	0.232	0.232	0.230
30	**0.270**	**0.278**	0.227	0.226	0.257	0.257	0.266

TABLE 14. Twitter Neg Region Entity Comparison

k	**10**	**20**	40	80	160	320	640
3	**0.253**	**0.247**	0.200	0.141	0.153	0.153	0.130
5	**0.238**	**0.247**	0.215	0.131	0.195	0.195	0.165
10	**0.238**	**0.243**	0.206	0.154	0.219	0.219	0.181
20	**0.247**	**0.248**	0.210	0.196	0.232	0.232	0.230
30	**0.270**	**0.278**	0.227	0.226	0.257	0.257	0.266

TABLE 15. Feature Importances of Age Models

Rank	**Twitter** 320	score	**Twitter Neg** 320	score	**Web** 320	score	**Web Neg** 320	score
1	do	0.952	given	0.418	**society**	0.538	**society**	0.589
2	behind	0.0281	(	0.2196	**student**	0.00918	**student**	0.0557
3	woman	0.0056	like	0.146	reserved	0.0388		0.0539
4	3	0.00429	.	0.0268		0.0287	reserved	0.0412
5	)	0.00244	center	0.023	copyright	0.0273	copyright	0.0327
6	exists	0.00147	3	0.0171	**person**	0.0127	**person**	0.0113
7	**person**	0.00146	/	0.0882	**work**	0.00833	**public**	0.00907
8	.	0.00123	**person**	0.00455	include	0.00739	woman	0.00707
9	become	0.00123	do	0.0045	**age**	0.00614	**drink**	0.00596
10	/	0.000742	Dazai	0.0043	**public**	0.00545	all	0.00502

to introduce some domain adaptation method, which is often used in transfer learning for correcting differences in the distribution of the features between both domains.

5. Conclusions

In this paper, we propose a method of retrieving posts on SNSs by specifying a topic query and an entity query. Our method estimates types of users by using a classifier whose input is the vocabulary in their posts and their profile data. We obtain datasets for training such classifier through the Web search by the keyword describing the target type of users, such as "student", instead of Twitter search with that keyword. It is because we expect that Web pages are longer than tweets and include more vocabulary describing types of people. Our experimental results show that the classifiers trained by the dataset consisting of Web pages outperforms the classifiers trained by the dataset consisting of tweets for many categories of entity queries although the datasets of Web pages have less vocabulary than the datasets of tweets. We also confirmed that a larger dataset does not always lead to a better query performance. It is because a larger dataset including low-ranked pages or tweets may include many noisy data. In future work, we plan to develop a method of automatically determining the optimal size of the dataset for a given entity query. We also plan to develop a method of domain adaptation by utilizing the difference of distribution of vocabulary between Web and Twitter.

References

[1] R. Nagmoti, A. Teredesai, and M. De Cock, "Ranking approaches for microblog search," in *Proc. of WI-IAT*, 2010, pp. 153–157.

[2] Z. Luo, M. Osborne, S. Petrovic, and T. Wang, "Improving twitter retrieval by exploiting structural information." in *AAAI*, 2012.

[3] M. Efron, "Hashtag retrieval in a microblogging environment," in *Proc. of SIGIR*, 2010, pp. 787–788.

[4] D. Metzler, C. Cai, and E. Hovy, "Structured event retrieval over microblog archives," in *Proc. of the 2012 Conference of the North American Chapter of the ACL*, 2012, pp. 646–655.

[5] C. H. Lau, Y. Li, and D. Tjondronegoro, "Microblog retrieval using topical features and query expansion," in *TREC*, 2011.

[6] R. Pochampally and V. Varma, "User context as a source of topic retrieval in twitter," in *Workshop on Enriching Information Retrieval (with ACM SIGIR)*, 2011, pp. 1–3.

[7] M. Albakour, C. Macdonald, I. Ounis *et al.*, "Identifying local events by using microblogs as social sensors," in *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, 2013, pp. 173–180.

[8] J. Lu, M. P. Kato, T. Yamamoto, and K. Tanaka, "Entity identification on microblogs by CRF model with adaptive dependency," *The IEICE Transactions*, pp. 2295–2305, 2016.

[9] D. M. Herzig and T. Tran, "Heterogeneous web data search using relevance-based on the fly data integration," in *WWW*, 2012, pp. 141–150.

[10] H. Halpin and V. Lavrenko, "Relevance feedback between web search and the semantic web," in *IJCAI*, 2011, pp. 2250–2255.

[11] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

[12] V. M. K. Peddinti and P. Chintalapoodi, "Domain adaptation in sentiment analysis of twitter." *Analyzing Microtext*, vol. 11, p. 05, 2011.

[13] D. Rieman, K. Jaidka, H. A. Schwartz, and L. Ungar, "Domain adaptation from user-level facebook models to county-level twitter predictions," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2017, pp. 764–773.

[14] J. J. Rocchio, "Relevance feedback in information retrieval," *In The Smart system-experiments in automatic document processing*, 1971.

[15] D. Kataoka, M. P. Kato, T. Yamamoto, O. Hiroaki, and K. Tanaka, "Context-aware relevance feedback over sns graph data," in *Proc. of WI*. ACM, 2017, pp. 823–830.

[16] S. Whiting, I. A. Klampanos, and J. M. Jose, "Temporal pseudo-relevance feedback in microblog retrieval," in *ECIR*, 2012, pp. 522–526.

[17] J. Choi and W. B. Croft, "Temporal models for microblogs," in *Proc. of CIKM*, 2012, pp. 2491–2494.

[18] T. Miyanishi, K. Seki, and K. Uehara, "Improving pseudo-relevance feedback via tweet selection," in *Proc. of CIKM*, 2013, pp. 439–448.

[19] F. Liang, R. Qiang, and J. Yang, "Exploiting real-time information retrieval in the microblogosphere," in *Proc. of JCDL*, 2012, pp. 267–276.

[20] S. Whiting, Y. Moshfeghi, and J. M. Jose, "Exploring term temporality for pseudo-relevance feedback," in *Proc. of SIGIR*, 2011, pp. 1245–1246.

[21] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[22] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM TOIS*, pp. 422–446, 2002.

[23] L. Breiman, *Classification and regression trees*. Routledge, 2017.

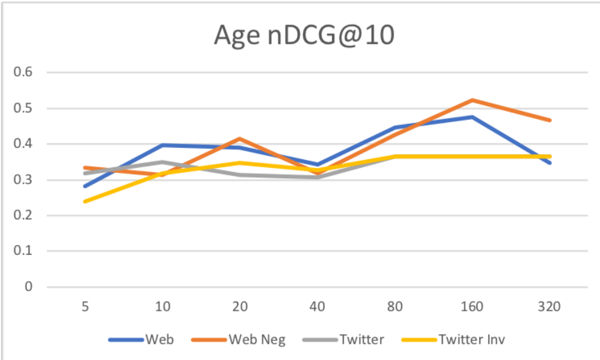

Figure 3. nDCG@10 Age Entity Comparison

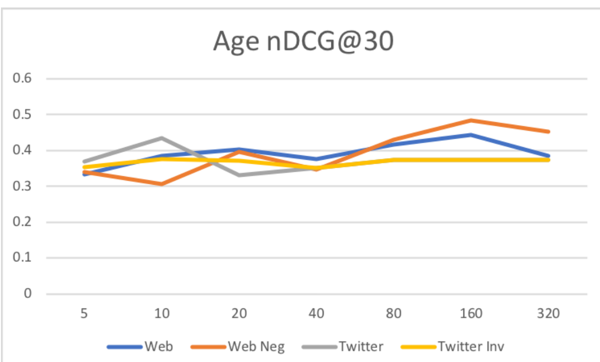

Figure 4. nDCG@30 Age Entity Comparison

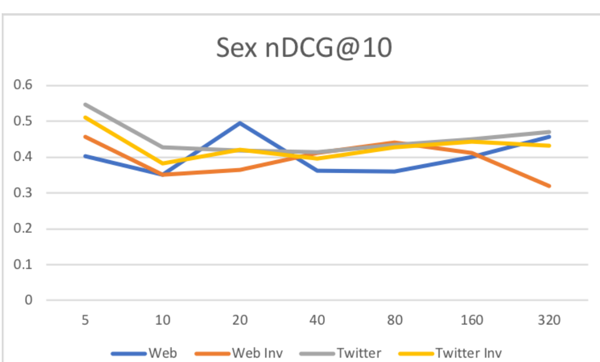

Figure 5. nDCG@10 Sex Entity Comparison

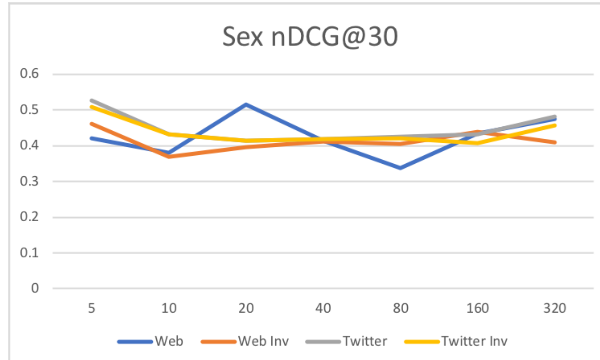

Figure 6. nDCG@30 Sex Entity Comparison

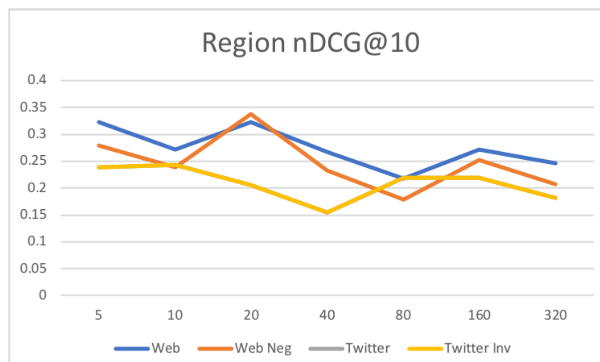

Figure 7. nDCG@10 Region Entity Comparison

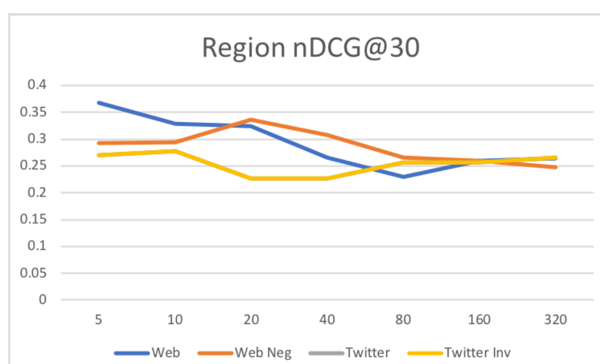

Figure 8. nDCG@30 Region Entity Comparison