Replacing Log-Based Profiles to Context Profiles and Its Application to Context-aware Document Clustering

YUSUKE HOSOI
Kyushu University
Department of Informatics
744 Moto-oka, 819 0395 Fukuoka
JAPAN
yusuke.hosoi@inf.kyushu-u.ac.jp

YUTA TANIGUCHI
Kyushu University
Department of Informatics
744 Moto-oka, 819 0395 Fukuoka
JAPAN
yuta.taniguchi@inf.kyushu-u.ac.jp

DAISUKE IKEDA
Kyushu University
Department of Informatics
744 Moto-oka, 819 0395 Fukuoka
JAPAN
daisuke@inf.kyushu-u.ac.jp

Abstract: As the number of documents is increasing rapidly, personalization is becoming more and more important to find desired documents efficiently. In typical personalization frameworks, a user profile created by its histories is necessary but it may include different contexts even for one user. In this paper, we develop a framework of personalization for information retrieval in various contexts. The main idea of this paper is twofold: firstly we use a vector as a generalized profile, called a *context profile*, of a user, a context, or a segment, and secondly we use corpora instead of user histories. This means that we can create a profile for a context at low cost and choose it according to contexts. Moreover, we can easily obtain virtual profiles from created profiles since profiles are just vectors. To evaluate the proposed framework, we have created many context profiles from a popular corpus, adjusted usual document vectors to contexts, and compared to adjusted document vectors and original ones. Effectiveness of adjusted documents are also confirmed by document clustering which creates different clusters according to contexts.

Key–Words: Information retrieval, Vector space model, Context-aware, Personalization, User profile, Term weighting, Document clustering

1 Introduction

Due to the receent rapid increase of documents, it is becoming difficult for users to find desired documents efficiently. Personalization is considered as one of the promising ways to support to do that. Since personalization is a general framework, it is applicable to many applications. In fact, we can see many applications, such as Web search [1, 2, 3, 4], document clustering [5], and recommendation [6].

In typical frameworks of personalization, user histories are required to accommodate user interests [7]. However, obtaining user histories is at high cost in general. Moreover, the cold start problem is a well-known problem of personalization based methods [8, 9]. Although these problems of user histories are serious, we focus on another problem of user histories, that is, the history data of only one user may include different contexts, such as work or hobby, because we search in many contexts but we do not explicitly give contexts to search systems. For example, consider the browser log of a college student at home. It might contain logs of searching topics related to homework, items of hobbies, and many other things.

In this paper, we replace a user profile by a *context profile*, which represents contexts. To do so, we abstract a profile as a vector of weights for terms, like a document vector in the vector space model. Then, the profile is used to modify a document vector so that terms with large weights in a context profile, which mean these terms are familiar to the user, are amplified in the resulting document vector and those of terms with small ones are filtered. Once we treat a user profile as a vector, we can use some corpus to create vectors as profiles. If the corpus is a set of a specific hobby, then we can see the obtained vector as the profile for this context.

The significance of the context profiles is twofold: Firstly we we can choose profiles independent to user histories. For example, consider that a user wants to find stories with some adventure taste although the user usually prefers to read books of science fiction. In this case, personalization methods based on the history fail to find desired documents while the user can choose an appropriate context profile and so is expected to find desired one in our framework. Secondly we obtain virtual context profiles from created profiles since now profiles are just vectors. So, for example, we can use a context profile of “adventure” or “mys-

tery" by sum of corresponding two profiles.

In the literatures, we can find many methods which use similar term-based profiles, such as [10, 7, 11]. However, these methods are firmly-fused with their applications while our framework is separated from both user histories and applications. This is also a significant point of the proposed generalized user model. Due to this, we can choose a profile among profiles created from corpora and thus we can obtain different results according to the chosen profile.

We have confirmed that we can create many profiles of contexts using a popular corpus and resulting vectors are modified according to profiles. For example, in the document vector of "Alice's Adventures in Wonderland" created by BM25 [12], a popular term weighting method, we find that characteristic but unpopular words, such as, "hookah" and "fish-footman", have large weights while we have different types of words whose weights are large when we use context profiles. For example, "kid", "bottle", and "cattle" have large weights in the adventure context, and "frighten" and "murder" in the mystery context. We have also confirmed profiles created from two or more different contexts. We see that words some words related to mystery are removed from words related to adventures when we use the context of "adventure" - "mystery".

We have also confirmed that we can use this framework for document clustering, by checking a cluster containing "Alice's Adventures in Wonderland". When we use BM25 as a term weighting method, cluster centroids contain too specific words, such as "dinah", "carroll", and "queer-shaped", as larger weights. When we use the profile of hobbies, we have more popular words, such as "mustard", "gymnastics", and "pepper". This means that obtained clusters are easy to interpret, thank to context profiles.

2 Proposed Method

In this section, we introduce context profiles after we explain document vectors. Then, we introduce context-aware document vectors.

2.1 Document Vectors

Suppose we have $\boldsymbol{D}$ of N documents. A document in $\boldsymbol{D}$ is represented by a set of words. In other words, we use the bag-of-words model [13].

To represent a document $\boldsymbol{d} \in \boldsymbol{D}$ as a document vector, suppose we have vocabulary of M terms. Then, a document vector $\boldsymbol{d}$[1] is defined as $\boldsymbol{d} = (w_1, w_2, \ldots, w_M)$, where w_i is a weight of the *i-th* term for the document. The weights are generally computed by term weighting methods, such as TF-IDF [14] or BM25 [12].

In this study, we use BM25 for term weighting to documents since it is known that BM25 shows the best performance for document representation. By using BM25, a weight $w_{i,j}$ of the *i-th* term for the *j-th* document $\boldsymbol{d}_j$ is formally computed as

$$w_{i,j} = \frac{tf_{i,j} \cdot (k_1 + 1)}{tf_{i,j} + k_1 \cdot (1 - b + b \cdot \frac{len(\boldsymbol{d}_j)}{avglen(\boldsymbol{D})})} \cdot idf_i,$$

where $tf_{i,j}$ is the term frequency of the *i-th* term in $\boldsymbol{d}_j$, $len(\boldsymbol{d}_j)$ is the length of the *j-th* document, the $avglen(\boldsymbol{D})$ is average the length of documents, and k_1 and b are parameters that takes $k_1 = 1.2$, $b = 0.75$. idf_i is defined as

$$idf_i = \log \frac{N}{df_i},$$

where N is the total number of documents and df_i is the document frequency of the *i-th* term.

2.2 Replacing of Log-Based Profiles

We introduce a context profile which represents a context as a vector. We use corpora for computing context profiles. For example, papers of informatics is used for the informatics-aware profile. In this profile, informatics terms, such as "clustering" and "tf-idf", have large weights. In contrast, cooking-aware terms, such as "allspice" and "meuniere", have small weights. In this way, we can use a profile which represents a single context.

Suppose we have the same vocabulary of M terms for document vectors. Then we define a context profile $\boldsymbol{p}$ as a vector $\boldsymbol{p} = (v_1, v_2, \ldots, v_M)$, where v_i is the value of the *i-th* term for the context. This profile is represented in the same vector space of document vectors.

In this study, we create context profiles by TF-IDF because our preliminary experiment showed that desired terms were given larger weights by TF-IDF than BM25. Suppose we have L documents. By using TF-IDF, we compute v_i of the *i-th* term by summing TF-IDF values by

$$v_i = \sum_{j=1}^{L} tf_{i,j} \cdot idf_i.$$

[1] We use the same symbol both for a document and a document vector.

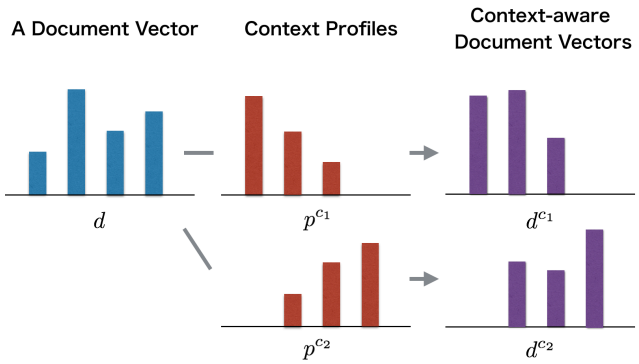

Figure 1: Overview of how to create context-aware document vectors $\boldsymbol{d}^c$. To adjust a document vector $\boldsymbol{d}$ to a context, we use a context profile $\boldsymbol{p}^c$.

2.3 Combination of Context Profiles

We can combine context profiles to obtain another context profile because context profiles are represented as vectors. We consider four types of combination : addition, subtraction, multiplication and division. Not only combination of two profiles but also combination of more profiles. For instance, we combine the adventure-aware profile with the mystery-aware profile into the adventure-and-mystery profile by adding profiles. Also, we combine the fiction-aware profile with mystery-aware profile into the fiction-without-mystery profile by subtracting profiles.

2.4 Context-aware Document Vectors

A context-aware document vector represent a document vector which adjusted to a context. For instance, a book of "Alice's Adventures in Wonderland" is adjusted to the context of adventure or mystery. To adjust a document vector $\boldsymbol{d}$ to a given context $\boldsymbol{p}^c$, we merge $\boldsymbol{p}^c$ and $\boldsymbol{d}$ into the context-aware document vector $\boldsymbol{d}^c$ (see Figure 1).

We can consider that functions of context profiles are filters and amplifiers to document vectors. In other words, a context profile can also be used to filter and amplify weights in document vectors to be suitable for this context. That is to say that if an information is appropriate to a context, the information will be amplified. Otherwise, the information will be removed or reduced.

In this study, we compute context-aware document vectors by multiplying each element of $\boldsymbol{d}$ and $\boldsymbol{p}^c$:

$$\begin{aligned} \boldsymbol{d}^c &= f(\boldsymbol{d}, \boldsymbol{p}^c), \\ w_i^c &= w_i \cdot v_i^c, \end{aligned}$$

where w_i is weight of the *i-th* term for $\boldsymbol{d}$ and v_i^c one of the *i-th* term for c.

3 Experiments

In this section, to confirm that, by context profiles, document vectors are changed according to context profiles, we show two types experiments: one is that we create the context-aware document vectors and check them directly, and the other is that we apply context-aware documents to a popular clustering tool and check obtained clusters indirectly.

3.1 Data

We use two corpora, the one for document vectors, and the other for context profiles.

To create document vectors, we used text data from Project Gutenberg[2]. We only used the English books in the top 100 rankings of downloads as of July 4, 2013. Books include "Alice's Adventures in Wonderland" and "The Adventures of Sherlock Holmes". The reason why we have chosen this corpus that it contains many books which contain various contexts, and thus we can expect to obtain books appropriate to given various context profiles.

To create context profiles, we used the Brown Corpus[3] which is a well known dataset consisting of various categories of documents. Brown corpus is divided into 15 categories and has 500 documents, each of which is constituted by more than 2000 words. Categories of documents range from "editorial" to "adventure". Thus, we can use various categories as various contexts.

In order to represent documents as the vector space model, we preprocessed documents with NLTK[4] library, which stands for Natural Language ToolKit: lowercasing, tokenization, using only content words (noun, adjective, verb and adverb), lemmatization and deleting stop words.

3.2 Creation of Context-aware Document Vectors

The purpose of the experiment is to confirm that document vectors are adjusted to a context. That's why we create many document vectors according to contexts. However, it is difficult to check all of the created documents. Thus, we integrate them into one representative vector, called an *summarized vector*, by summing document vectors.

[2]http://www.gutenberg.org/
[3]http://khnt.aksis.uib.no/icame/manuals/brown/
[4]http://nltk.org/

Table 1: 22 contexts and its number. There are 1 of no context case (none) and 15 of one context case and 6 of combined context case. The context of imaginative is combined adventure, fiction, humor, mystery, romance and science fiction. The context of informative is combined belles lettres, editorial, government, hobbies, learned, lore, news, religion and reviews.

number	context	number	context
00	none	11	news
01	adventure	12	religion
02	belles_lettres	13	reviews
03	editorial	14	romance
04	fiction	15	science_fiction
05	government	16	adventure+government
06	hobbies	17	adventure-fiction
07	humor	18	adventure-government
08	learned	19	adventure-mystery
09	lore	20	imaginative
10	mystery	21	informative

Formally, a summarized vector is created as follows. For a context c, the summarized vector of context-aware document vectors $\boldsymbol{d}_j^c$ is calculated as

$$summarized\ vector^c = \sum_{j=1}^{N} \boldsymbol{d}_j^c, \qquad (1)$$

where we used only top 100 terms of each document in the descending order of weights because we think them as informative terms.

We created context-aware document vectors as follows. First, we computed document vectors from 96 documents of Project Gutenberg. Second, we computed context profiles from 500 documents of Brown Corpus. We applied TF-IDF to 500 documents, then we created 15 context profiles by combining documents in each category. Let c be the context c and its profile $\boldsymbol{p}^c$. We defined context value $v_i^c \in p^c$ of the *i-th* term in the context c as

$$v_i^c \quad = \quad \sum_{j \in c} tf_{i,j} \cdot idf_i.$$

Third, we create 6 combined profiles by combining 15 single contexts profiles. Thus we use the total of 22 profiles as Table1, including the case that no context is given. Finally, we created context-aware document vectors. We created a context-aware document vector $\boldsymbol{d}_j^c$ by multiplying each element of $\boldsymbol{d}_j$ and $\boldsymbol{p}^c$.

Table 2 shows term weights rankings of several contexts and BM25 in "Alice's Adventures in Wonderland". These tables consists of six columns, each of which corresponds a context and shows top 15 terms in descending order of weights in this context. The label of each column denotes the context, such as adventure. Please note that the label of each column is combination of a term weighting method (BM25) and a context, but we abbreviate description of a term weighting method (BM25). We see that terms with large weights are changed according to each context, and the change is suitable for the each context. For instance, the context of mystery puts large weights to terms, such as "frighten" and "murder", which seems to be suitable for this context. However, the case of no context (none) puts large weights to too specific terms, such as "hookah" and "fish-footman". Moreover, combined profiles change suitably document vectors. In the case of adventure + government, terms are related to adventure and government. In the case of adventure - fiction, terms are emphasized the context of adventure. We consider result of this, many general fiction terms are included from context of fiction. Adventure is subtracted general fiction terms, so adventure terms are emphasized. In the case of adventure - government, terms are similar to a single profile of adventure. Because, government is not related to adventure. Even subtracting the element does not matter from the adventure, there is no change in the top words of ranking. In the case of adventure - mystery, terms, such as "killing", are gone from the ranking. In this way, combination of context profiles also performs role of the filter. In the case of both imaginative and informative, terms are suitable for context.

Figure 2 shows 22 bar graphs of relevance of a single context in summarized vectors of 22 contexts, including the case that no context is given. In each bar graph, horizontal axis corresponds to a single context and vertical axis their relevance. We compute rele-

Table 2: Term weights rankings of several contexts and BM25 in "Alice's Adventures in Wonderland". These tables consists of six columns, each of which corresponds a context and shows top 15 terms in descending order of weights in this context. The label of each column denotes the context, such as adventure. Please note that the label of each column is combination of a term weighting method (BM25) and a context, but we abbreviate description of a term weighting method (BM25).

none		adventure		fiction		government		hobbies		mystery	
oop	9.629	maybe	0.853	ma	1.802	planning	0.514	mustard	0.844	maybe	0.821
lory	9.629	yelled	0.791	kid	1.340	adoption	0.247	pepper	0.481	kid	0.616
soo	9.629	grinned	0.595	schoolroom	0.687	pool	0.246	trot	0.351	tunnel	0.605
hookah	9.473	kid	0.542	scratching	0.606	in.	0.214	cattle	0.344	grinned	0.541
jury-box	9.341	bottle	0.496	upstairs	0.552	protection	0.197	tougher	0.296	spade	0.471
ootiful	9.341	straightened	0.452	maybe	0.484	speaker	0.187	planning	0.271	right-hand	0.457
rabbit-hole	9.341	pool	0.444	crouched	0.480	machine	0.178	multiplication	0.255	straightened	0.441
muchness	9.129	grunted	0.405	toffee	0.461	resource	0.172	hunting	0.253	frighten	0.402
eaglet	9.129	cackled	0.390	kitchen	0.458	sh	0.167	gallon	0.228	nodded	0.389
fish-footman	8.732	cattle	0.389	powdered	0.409	encourage	0.120	saucepan	0.214	murder	0.383
dinn	8.732	livery	0.384	butter	0.407	carrier	0.115	trim	0.209	checked	0.350
uglification	8.732	rustling	0.327	funny	0.393	grant	0.112	fun	0.207	stair	0.331
mercia	8.732	hall	0.321	grinned	0.393	encouraging	0.111	butter	0.204	toast	0.326
barrowful	8.732	killing	0.320	crawling	0.392	personal	0.111	sugar	0.189	paused	0.323
caucus-race	8.732	nodded	0.306	stair	0.386	teaching	0.105	jar	0.167	walked	0.319

adventure+government		adventure-fiction		adventure-government		adventure-mystery		imaginative		informative	
planning	0.514	yelled	1.243	maybe	0.883	yelled	1.071	maybe	0.672	axis	0.263
pool	0.443	maybe	0.978	yelled	0.844	pool	0.611	kid	0.543	planning	0.177
maybe	0.390	straightened	0.720	grinned	0.635	cackled	0.572	grinned	0.464	vote	0.150
yelled	0.351	livery	0.714	kid	0.578	cattle	0.570	funny	0.365	cattle	0.148
grinned	0.264	bottle	0.673	bottle	0.529	livery	0.564	ma	0.347	education	0.141
adoption	0.247	pool	0.618	straightened	0.482	bottle	0.531	yelled	0.320	jury	0.123
kid	0.241	grinned	0.614	grunted	0.432	rustling	0.480	straightened	0.291	mustard	0.122
protection	0.228	cattle	0.601	cackled	0.416	rattling	0.438	bottle	0.248	atom	0.105
bottle	0.220	grunted	0.564	cattle	0.414	dodged	0.418	nodded	0.240	machine	0.096
in.	0.214	rattling	0.555	livery	0.410	hoarsely	0.404	kitchen	0.235	club	0.095
straightened	0.201	dodged	0.529	rustling	0.349	ax	0.387	crazy	0.226	civil	0.093
speaker	0.187	hoarsely	0.511	killing	0.341	maybe	0.368	worried	0.218	emphasis	0.093
resource	0.181	jury	0.482	nodded	0.326	dive	0.365	walked	0.214	series	0.087
grunted	0.180	crawled	0.477	rattling	0.319	rustled	0.338	baby	0.208	bill	0.086
machine	0.178	killing	0.475	hall	0.313	peg	0.338	pink	0.208	involved	0.084

vance by summing of the weights of top 1000 terms in a single context from each summarized vector. Numbers which correspond to the context of summarized vectors are shown on the right of the figure. In the case of no context (none), relevance to many context are high. However, in the case of a single context, relevance to one context is only high. In the combined contexts by addition (no. 16, 20, 21), relevance to contexts of the element for addition are high. In the combined contexts by subtraction (no. 17, 18, 19), relevance to context of right term to subtraction is low and relevance to context of left term to subtraction is high.

Thus, we conclude that context profiles can suitably adjust document vectors according to a context. Also, we can choose various context by combining of contexts profiles.

3.3 Application to Context-awere Document Clustering

The purpose of the experiment is confirmation that context-aware document vectors apply to document clustering. That's why we create many clusters according to contexts. However, it is difficult to check all of the created clusters. Thus, we integrate them into one representative vector, called an *summarized centroids*, by summing cluster centroids. Please note difference between previous summarized centroids, shown in equation (1), and these summarized centroids. Previous summarized vectors are summation of context-aware *document vectors*, and these summa-

Figure 2: 22 bar graphs of relevance of a single context in summarized vectors of 22 contexts, including the case that no context is given. In each bar graph, horizontal axis corresponds to a single context and vertical axis their relevance. We compute relevance by summing of the weights of top 1000 terms in a single context from each summarized vector. Numbers which correspond to the context of summarized vectors are shown on the right of the figure. In each context, relevance to contexts are changed suitably for the context.

```
【BM25--none】
 Item num : 3
 Item examples :
   Through the Looking-Glass, Grimms' Fairy Tales
 Cluster centroids :
   dinah:0.105538, carroll:0.102633, together.:0.101588, queer-shaped:0.101412

【BM25--hobbies】
 Item num : 6
 Item examples :
   Through the Looking-Glass, The Yellow Wallpaper, 2 B R 0 2 B
 Cluster centroids :
   mustard:0.243421, gymnastics:0.218733, pepper:0.193693, glue:0.185022, lumber:0.173075

【BM25--science fiction】
 Item num : 5
 Item examples :
   Through the Looking-Glass, A Doll's House, Metamorphosis, The Wonderful Wizard of Oz
 Cluster centroids :
   maybe:0.320334, politely:0.169769, smiled:0.165124, needle:0.162478, jumping:0.157007

【BM25--adventure+government】
 Item num : 3
 Item examples :
   Through the Looking-Glass, The Wonderful Wizard of Oz
 Cluster centroids :
   maybe:0.250999, yelled:0.215292, bottle:0.192539, grinned:0.181056, truck:0.177693

【BM25--adventure-fiction】
 Item num : 7
 Item examples :
   The Mysterious Affair at Styles, Secret Adversary, A Christmas Carol
 Cluster centroids :
   maybe:0.383684, anyway:0.266437, let's:0.263131, bottle:0.17055, thoughtfully:0.142197

【BM25--imaginative】
 Item num : 10
 Item examples :
   Through the Looking-Glass, Metamorphosis, The Wonderful Wizard of Oz
 Cluster centroids :
   maybe:0.303255, let's:0.290984, funny:0.196652, anyway:0.18306, stair:0.161097

【BM25--informative】
 Item num : 5
 Item examples :
   Through the Looking-Glass, Grimms' Fairy Tales, The Wonderful Wizard of Oz
 Cluster centroids :
   radiation:0.242308, dictionary:0.176554, center:0.16039, cultural:0.144646
```

Figure 3: This shows several clusters obtained by clustering usual document vectors (BM25) and those modified by some context profiles, such as BM25–hobbies or BM25–science fiction, where the number of items in each cluster and some of their titles are given in "Item num" and "Item examples", and cluster centroids are high weight terms which have great effect for clustering. We have chosen clusters containing "Alice's Adventures in Wonderland" to compare them from the same viewpoint.

Figure 4: 22 bar graphs of relevance of a single context in summarized centroids of 22 contexts, including the case that no context is given. In each bar graph, horizontal axis corresponds to a single context and vertical axis their relevance. We compute relevance by summing of the weights of top 1000 terms in a single context from each summarized vector. Numbers which correspond to the context of summarized centroids are shown on the right of the figure. In each context, relevance to contexts are changed suitably for the context.

rized centroids are summation of context-aware *cluster centroids*.

Formally, an summarized centroid is created as follows. For a context c, the summarized centroid of context-aware cluster centroids $\boldsymbol{c}_j^c$ is calculated as

$$summarized\ centroids^c = \sum_j \boldsymbol{c}_j^c,$$

where $\boldsymbol{c}_j^c$ is cluster centroids of the *j-th* cluster in the context c. where we used only top 100 terms of each cluster centroids in descending order of weights because we think them as informative terms.

For application to clustering, we used top 100 terms in descending order of weights by each document. Also, we used bayon [5] of a clustering tool. We used clustering by assigning limit value of cluster bisection without assigning the number of clusters.

Figure 3 shows several clusters obtained by clustering usual document vectors (BM25) and those modified by some context profiles, such as BM25–hobbies or BM25–science fiction, where the number of items in each cluster and some of their titles are given in "Item num" and "Item examples", and cluster centroids are high weight terms which have great effect for clustering. We have chosen clusters containing "Alice's Adventures in Wonderland" to compare them from the same viewpoint. We see that items and cluster centroids are changed according to each context, and it change is suitable for the each context. For instance, cluster centroids of the context of hobbies are changed into cooking terms such as "mustard" and "pepper". Also, items of the context of science fiction is changed such as "Metamorphosis". We consider that these changes are suitable for the each context. However we consider that, in the case of no context (BM25–none), terms of the cluster centroids are too specific such as "dinah" and "carroll", and the number of items are too small.

Figure 4 shows 22 bar graphs of relevance of a single context in summarized centroids of 22 contexts, including the case that no context is given. In each bar graph, horizontal axis corresponds to a single context and vertical axis their relevance. Numbers which correspond to the context of summarized centroids are shown on the right of the figure. In each context, relevance to contexts are similar to previous result of relevance of summarized vectors. Thus, clustering result is changed suitably for the context by context profiles.

The results, we consider as follows. We consider that change of cluster centroids are change of clustering. Since, cluster centroids are determining factor of clustering. Thus, we conclude that context profiles can suitably adjust results of clustering to a context.

[5] https://code.google.com/p/bayon/

4 Conclusion

To provide more appropriate profiles to information retrieval systems, we proposed context profiles and context-aware document vectors. The context profiles are made from arbitrary corpora or combining context profiles and they are not necessarily related to a particular user, but to a segment of users or to some context of documents. The context-aware document vectors are computed from those context profiles and usual document vectors represented in the vector space model. In our experiment, we made many context profiles from the Brown Corpus for each category, and then we performed document clustering taking account of each context. Comparing each result of the context-aware document vector, we found that each vector is changed according to the context. Comparing each result of the clustering, we also found that each result of clustering is changed according to the context.

Therefore, we conclude that context profiles can suitably adjust document vectors and results of clustering to a context. Also, a user can choose arbitrarily the context.

Important future works include the following:

- Quantitative evaluation : In this study, we show only qualitative evaluation. We will examine a dataset for our methods evaluation.
- Combination of a context profile and a document vector : We will examine more various combinations.
- Applications to other information technologies: There are many information technologies which use the vector space model. We will examine application to other information technologies.

Acknowledgment

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B), Number 24300059.

References:

[1] Nicolaas Matthijs and Filip Radlinski. Personalizing Web search using long term browsing history. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*, pages 25–34, New York, NY, USA, 2011. ACM.

[2] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th International Conference on World Wide Web*, pages 727–736, New York, NY, USA, 2006. ACM.

[3] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive Web search based on user profile constructed without any effort from users. In *Proceedings of the 13th International Conference on World Wide Web*, pages 675–684, New York, NY, USA, 2004. ACM.

[4] Jaime Teevan, Meredith Ringel Morris, and Steve Bush. Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 15–24, New York, NY, USA, 2009. ACM.

[5] Chih-Ping Wei, Roger Chiang, and Chia-Chen Wu. Accommodating individual preferences in the categorization of documents: A personalized clustering approach. *J. Manage. Inf. Syst.*, 23(2):173–201, October 2006.

[6] Liliana Ardissono, Cristina Gena, Pietro Torasso, Fabio Bellifemine, Angelo Difino, and Barbara Negro. User modeling and recommendation techniques for personalized electronic program guides. In *Personalized Digital Television - Targeting Programs to Individual Viewers, volume 6 of Human-Computer Interaction Series, chapter 1*, pages 3–26. Kluwer Academic Publishers, 2004.

[7] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. The adaptive Web. chapter User profiles for personalized information access, pages 54–89. Springer-Verlag, Berlin, Heidelberg, 2007.

[8] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pages 208–211, 2008.

[9] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, New York, NY, USA, 2002. ACM.

[10] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Semantic enrichment of twitter posts for user profile construction on the social Web. In *Proceedings of the 8th Extended Semantic Web Conference on the Semanic Web: Research and Applications - Volume Part II*, pages 375–389, Berlin, Heidelberg, 2011. Springer-Verlag.

[11] Fang Liu, Clement Yu, and Weiyi Meng. Personalized Web search by mapping user queries to categories. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 558–565, New York, NY, USA, 2002. ACM.

[12] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pages 42–49, New York, NY, USA, 2004. ACM.

[13] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[14] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317, October 1957.