

# A Query Randomizing Technique for breaking 'Filter Bubble'

Sangdon Joo\*, Sukyung Seo\*\*, Youngmi Yoon\*\*\*

## Abstract

The personalized search algorithm is a search system that analyzes the user's IP, cookies, log data, and search history to recommend the desired information. As a result, users are isolated in the information frame recommended by the algorithm. This is called 'Filter bubble' phenomenon. Most of the personalized data can be deleted or changed by the user, but data stored in the service provider's server is difficult to access. This study suggests a way to neutralize personalization by keeping on sending random query words. This is to confuse the data accumulated in the server while performing search activities with words that are not related to the user. We have analyzed the rank change of the URL while conducting the search activity with 500 random query words once using the personalized account as the experimental group. To prove the effect, we set up a new account and set it as a control. We then searched the same set of queries with these two accounts, stored the URL data, and scored the rank variation. The URLs ranked on the upper page are weighted more than the lower-ranked URLs. At the beginning of the experiment, the difference between the scores of the two accounts was insignificant. As experiments continue, the number of random query words accumulated in the server increases and results show meaningful difference.

▶ Keyword: Query, Bubble Filter, Randomization, History

## 1. Introduction

Pariser[1]가 주장한 'Filter bubble'은 사람들이 사용하는 검색 엔진의 개인화가 사용자의 정보를 차단하는 것을 말한다. 즉 Google, Yahoo, Bing 등의 검색 엔진들이 사용자의 정보를 수집하고, 이 수집한 정보를 바탕으로 각각의 알고리즘을 통해 개인화 검색 서비스를 제공함으로써 특정한 정보를 우선적으로 사용자에게 노출시킨다. 이에 대한 예로 취미가 여행이고 다른 나라를 자주 왕래한 사람이 '이집트'를 검색하였을 때, 이집트의 관광지, 호텔, 음식 정보 등이 상위 검색 결과에 랭크되어 노출된다. 하지만 이집트의 정치 상황, 경제 문제에 대한 검색 결과는 매우 하위 페이지에 나타나거나 제공하지 않는다. 이러한 서비스는 각 개인들의 관심사와 근접한 정보들을 먼저 보여주는 편의를 제공하나, 정보 제공의 다양성 면에서는 제한적이다.

검색 결과의 순위가 사용자에게 미치는 영향은 매우 크다. 사용자

는 검색 결과의 1, 2페이지를 보고 원하는 정보가 없으면 다른 검색을 시도하는 패턴을 보인다. 따라서 하위에 랭크된 검색 결과들은 접근성이 떨어진다. 즉 검색 엔진이 사용자의 정보를 수집하여 사용자의 관심사에 맞춘 정보들만 상위 페이지를 채운다면, 필요한 다른 정보들은 얻지 못하는 경우가 발생하게 된다.

또한, 이 검색 엔진들은 개인 정보를 광고 업체에게 제공하고 히스토리 데이터를 분석하여 검색어와 관련된 광고를 브라우저 상 배너로 띄우기도 한다. 컴퓨터 프로그래밍 언어에 대해서 검색을 하면 결과 옆 배너 광고로 '프로그래머 자격증 학원', '컴퓨터 관련 취업패키지' 등이 뜨는 것이 그 예이다.

이렇듯 개인화 검색은 이점과 단점이 공존한다. 그래서 이를 극복하기 위한 노력으로 Nagulendra[2] 등은 개인화에 갇힌 상황을 시각화하여 사용자에게 경각심을 알려주기 위한 도구를

\*First Author: Sangdon Joo, Corresponding Author: Youngmi Yoon

\*\*Sangdon Joo (sangdonjoo8@gmail.com), Dept. of Computer Engineering, Gachon University

\*\*\*Sukyung Seo (sukyung1229@gmail.com), Dept. of Computer Engineering, Gachon University

\*\*\*Youngmi Yoon (ymyoon@gachon.ac.kr), Dept. of Computer Engineering, Gachon University

• Received: 2017. 10. 21, Revised: 2017. 11. 07, Accepted: 2017. 12. 04.

개발하였다. 사용자의 계정이 특정 카테고리에 개인화된 것을 사람(계정)과 거품(카테고리) 형태로 나타내고 그 거품에 갇혀 있는 것을 표시한다. 이는 개인화를 극복하기 위해 인식론적으로 접근한 방식이다. 따라서 사용자의 개인 정보의 쓰임에 대한 경각심을 일깨워주는 도구로서 작용할 수 있으나, 사용자가 이 문제에 대해 지속해서 개인 정보를 관리하는 노력이 필요하다. 또한, 사용자가 관리 가능한 정보 외 검색 엔진에 축적된 정보에 대해서는 관리할 방법을 제시하지 못한 한계가 있었다.

Xing[3] 등은 사용자의 IP를 기반으로 단어를 검색했을 때의 검색 결과와 다른 국적 IP를 가진 계정에서 검색한 결과를 보여주고 사용자의 결과 URL을 비교하여 찾아볼 수 없었던 정보를 보여주는 형식의 접근을 하였다. 그러나 사전에 정보 제공의 동의를 받은 몇 개의 계정을 사용하여 누락된 결과를 보여주는 것에 국한되어 있었다. 또한, 각각의 계정의 히스토리 데이터가 쌓이면서 또 다른 개인화가 이루어지는 면이 있어서, 근본적인 개인화의 문제점을 사용자 개인의 계정 내에서 해결하기에는 한계가 있었다.

이 개인화 검색 알고리즘에서 사용되는 데이터의 종류로는 사용자의 접속 위치, IP 주소, 로그 데이터, 캐시, 방문 기록, 검색어, 페이지 접속 시간 등 많은 요소가 있지만, 본 연구에서는 이들 요소 중에서 가장 영향력 있는 데이터인 검색어를 타겟으로 선정하고, 이를 축적하여 개인화 알고리즘에 혼란을 야기하고자한다.

사용자가 사용한 검색어는 개인의 브라우저 내에 저장되고, 또한 서버로 전송하여 저장된다. 개인의 PC상에 저장되는 데이터들뿐만 개인화가 이루어진다면 이 데이터를 삭제하기만 하면 될 것이다. 그러나 동시에 서버에서도 이 계정의 검색 활동을 기록하기 때문에 이는 개인이 해결할 수 있는 범위를 벗어난다. 이에 서버에 저장되는 검색어를 무작위로 전송하여 개인화 알고리즘을 무력화시키는 방법을 제안한다. 이를 ‘검색어 랜더마이징’이라 명명한다.

랜더마이징을 위한 검색 엔진의 타겟으로 ‘구글(Google)’을 정하였고, 파이썬의 셀레니움(Selenium)[4] 라이브러리로 테스트의 자동화를 구축하였다. 파이썬의 RandomWord 라이브러리를 활용하여 단어들의 집합에서 무작위로 하나의 단어를 추출하고, 이를 검색을 위한 데이터로 실험을 진행하였다. 자동화된 검색 활동 후 URL의 변동을 분석하기 위해 정치&사회, 과학&기술, 스포츠, 금융, 이슈 각각의 분야에서 선정한 기준 검색어 100개를 개인화되어있는 계정과 개인화되지 않은 새로운 계정으로 검색 작업을 수행하고 검색 결과 URL을 스크래핑하여 분석 데이터로 사용하였다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 개인화 검색, URL 스크래핑에 대하여 기술하고 3장에서는 제안한 검색어 랜더마이징을 위한 데이터 수집, 스코어링에 대해 기술한다. 4장에서는 실험 결과 및 분석으로 실험 환경 및 데이터, 5장에서는 향후 이 실험에 관한 연구 방향과 논문에 대한 결론을 맺는다.

## II. Preliminaries

### 2. Related works

#### 2.1 Google's Personalized Search

구글은 사용자에게 맞춘 개인화 검색 결과를 보여주기 위한 연구를 거듭해 왔다. 이를 위해 사용자의 개인 정보를 수집한다. 대표적으로 위치, 검색 기록, 웹 기록, 소셜 네트워크의 데이터를 수집하고, 이뿐만 아니라 사용자의 검색 활동에 대한 데이터도 수집한다. 예로 하나의 검색어를 입력하고 사용자가 원하는 URL을 클릭하여 그 정보를 취할 때 그 특정한 사이트에 머문 시간, 사이트 내에 다른 URL로 이동한 경로 데이터들도 수집한다. 그 후 이 데이터를 분석하여 다음 검색 활동을 할 때 연관된 URL의 페이지 순위를 올리는 데 적용한다.

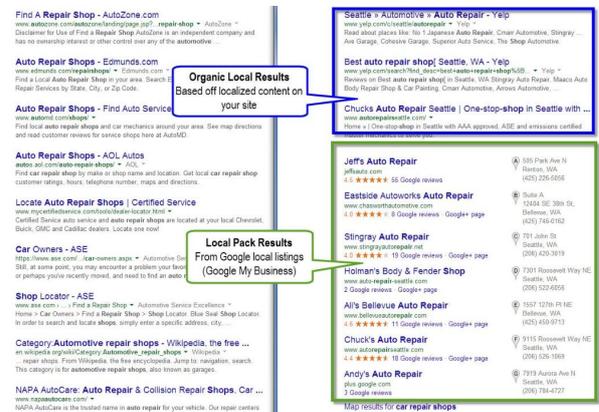


Fig. 1. Google Search Result Difference Between Login / Non Login



Fig. 2. Difference in Google Search Results as Search Terms Accumulate

Colborn[5] 등은 이 페이지 순위를 결정하는데 영향력 있는 요소를 계정 로그인, IP, 이전 검색어 기록이라 주장하였다. 그림 1은 요소 중 비 로그인인 된 상태에서의 검색 결과는 로그인되었을 때와 비교하여 이전 검색어와의 연관성이 적었다는 것 보여준다. 또한, 이 연구를 통하여 이전 검색어 기록을 수집하여 이 데이터가 다음 검색 활동에 영향을 주는 것이 확인되었다. 예로 그림 2는 ‘Javascript’를 검색한 후 ‘Book’을 검색한 결과 하위 URL에 Javascript 관련 서적의 URL이 순위에 포함되는 것을 확인하였다.

### 2.2 URL Scraping

스크래핑(Scraping)이란 HTML 문서 내의 데이터를 수집하는 기법으로써, 흔히 '크롤링'이라 알려져 있다. 이 스크래핑은 특정 URL에 접속해, 지정한 데이터를 가져와 다른 용도로 활용할 수 있다. 스크래핑을 위하여 파이썬 기반의 셀레니움(Selenium) 프레임 워크를 사용하였다. 그림 3은 셀레니움의 프로세스를 나타낸다.

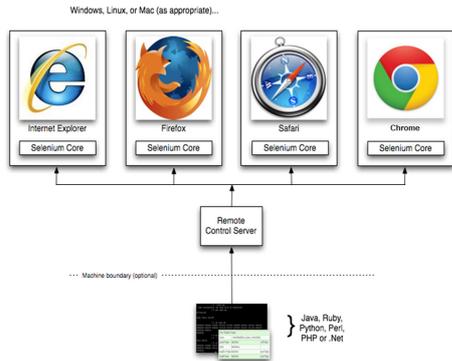


Fig. 3. Process of Selenium

이 셀레니움은 WebDriver를 이용해 여러 웹 브라우저 테스트를 실행해주도록 해주는 서버로서 검색 작업의 자동화를 지원하고, 작업 속에서 데이터를 추출하여 저장하는데 용이하다. 셀레니움은 HTML 문서의 Tag 속성을 확인하여 값을 변경하거나 태그를 클릭하는 등의 사용자 행동을 파이썬 함수로 표현하여 대신 수행한다. 하지만 웹 브라우저에서 Javascript를 구동시키는 방식으로 동작하기 때문에 웹 브라우저의 보안 모델을 넘어서는 테스트 (크로스 도메인, 파일 업로드 등)에는 제한적이다. 따라서 테스트 중 오토 봇으로 인식하고 이를 차단하는 기능인 Captcha를 무력화시키기에는 역부족이다. 이에 대해 본 연구에서는 Timesleep() 함수와 Random() 함수의 결합을 활

용하였다. 기존 검색어에 검색결과 저장을 완료한 후, 다음 검색활동을 시작하기 전까지의 시간 간격을 무작위로 지연시켜 오토 봇 인식문제 개선하는 방법을 사용하여 보완하였다.

### 2.3 The Effects of Search Engine Credibility and Information Ranking on Search Behavior

오지연[5] 등은 '검색 엔진이 제시하는 정보의 순위를 신뢰하며 검색 엔진이 정확하고 공정한 정보를 제공할 것이라고 믿는 경향이 있다.'라는 가정 하에 실험을 진행하였다. 이에 참가자들은 공신력이 높다고 인식된 '구글' 검색 엔진을 가상의 검색 엔진인 '네스폰'보다 신뢰 하였고, 동일한 뉴스를 읽었음에도 불구하고 그 뉴스가 검색 엔진에 의해 1위로 추천된 경우에 마지막 순위로 추천된 경우보다 해당 뉴스를 더 오랜 시간 읽었으며, 그 뉴스의 정보를 더 많이 활용했다고 답하였다. 이에 검색 엔진에 의해 1순위로 추천된 정보들이 낮은 순위의 정보들보다 사용자에게 끼치는 영향력이 더 큰 것을 고려하여 검색어 순위에 따른 URL 변동의 추가적으 것으로 예상하였다. 따라서 본 연구에서는 수집한 URL 데이터의 순위에 따라 가중치를 부여하여 스코어링 하였다.

## III. Method

### 3.1 System Overview

이 검색어 랜더마이징의 증명을 위하여 본 연구에서는 랜더마이징 후 매 회마다 URL 데이터를 수집한다. 개인화 검색 엔진의 혼란을 주기 위한 타겟으로는 구글 검색 엔진을 정하였고, 개인화된 계정 실험군(Personalized Account)과의 비교분석을

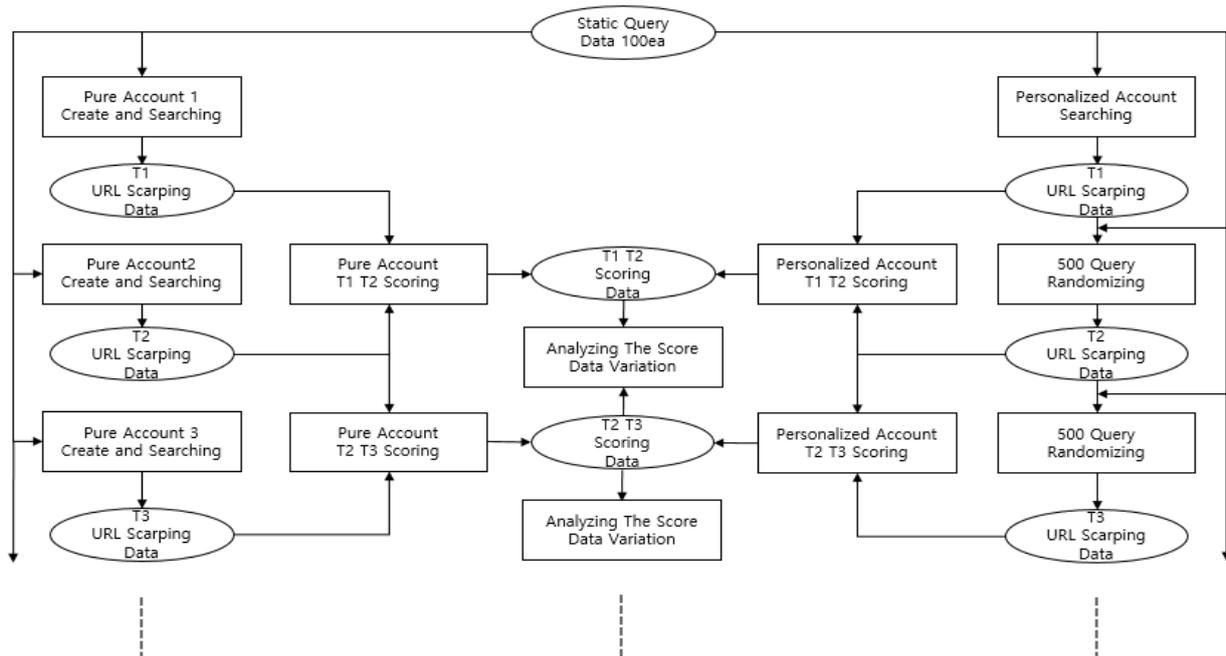


Fig. 4. Query Randomizing Process

위하여 새 계정을 생성하여 개인화되지 않은 상태의 대조군 (Pure Account)을 형성하여 실험하였다.

제한된 시스템의 전체적인 구조는 그림 4와 같다. 기준이 되는 100개의 검색어를 선정하고 이를 사용하여 실험군과 대조군 계정으로 검색을 한다. 기준 검색어를 검색할 때마다 결과 값으로 나오는 100개의 URL을 스크래핑하여 Excel 데이터로 저장한다. 이 데이터는 상위 URL부터 순차적으로 저장되기에 순위가 반영된다. 이렇게 검색어에 대한 데이터의 집합 T1이 생성 되면 실험군에 무작위 검색어 500개를 검색하는 검색어 랜더마이징(Query Randomizing)을 진행한 후 결과 URL을 수집하는 활동을 반복한다. 이렇게 T1과 T2의 URL 순위 데이터 세트가 준비되면 각각 검색어에 대한 URL의 순위 변동 차이를 분석하고 이를 스코어링 하였다. 이를 T1T2S(Score Change in between Time T1 and Time T2)라 한다. 이 스코어는 URL의 변동이 많이 이루어질수록 높게 나온다. T의 횟수가 늘어남에 따라 랜더마이징이 거듭되면서 실험군과 대조군의 스코어의 변화를 분석하였다.

### 3.2 Collect Experimental Data

#### 3.2.1 Select Static Query

Table 1. Static Query Set

Politics			
Loan	CEO	Industries	Technology
Money	Alibaba	North korea	Congressman
GOP	State	Demonstration	Response
president	Immigrant	Public	Brexit
Trump	Ravage	Opposition	Official
Tech & science			
Iron man	Internet	Google	Fingerprints
Ransomware	Cable	Wireless	TED
Developers	Design	Privacy	Zuckerberg
Data	Service	Production	AI
Android	Conference	Web	Drone
Sports			
Champ	Sprinter	PGA	Warriors
Number	Record	Round	Run
Boston	Deal	F1	Super bowl
Cubs	Injury	Championship	Rumor
Sign	Nike	Stage	Expire
Finance			
Stocks	Arab	Income	Routes
Quality	Sales	Toward	Gold
Investors	Portfolios	Transocean	Global Drive
Tax	Diversification	Micron	Biotech
Materials	Programmer	Ford	IRA
News			
Investigation	Cruise Ships	Starbucks	Employee
Greenhouse	Forces	Stronghold	Demands
Drug	Taxi	Manchester	Terror
Gains	Surgery	Secretary	Raping
Putin	Timeline	Fact	Syria

실험군과 대조군의 검색 활동에 사용되고, 검색 결과 URL 데이터 수집을 위한 검색어는 변동이 없어야 한다. 랜더마이징을 진행한 후 비교를 하기 위함이다. 따라서 기준이 되는 검색어를 포털 사이트인 'Yahoo' 내의 카테고리를 이용하였다[7]. 정치&사회, 과학&기술, 스포츠, 금융, 이슈, 총 5가지 분야에서 각각 20개의 단어를 선정하였고 표 1은 선정된 기준 검색어들의 집합이다. 선정된 시점부터 실험이 완료될 때까지 이 검색어는 실험/대조군의 검색 결과 URL 데이터를 수집하는 목적으로 매회 마다 동일하게 사용되었다.

#### 3.2.2 Control Group

대조군의 프로세스는 그림 5와 같다.

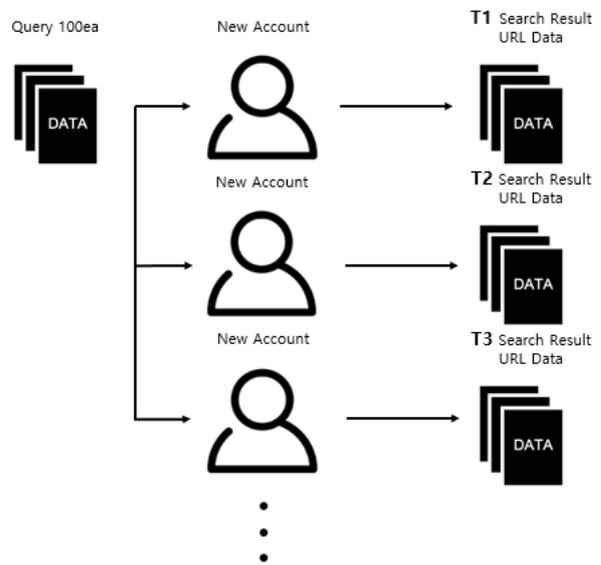


Fig. 5. Procedure of Control Group Search Activity

개인화가 되지 않은 결과 데이터를 수집하기 위하여 시간 T(Time-Term) 마다 새로운 계정을 생성하고 브라우저 내의 로그, 쿠키 데이터를 삭제하였다. 실험의 목적이 사용자가 제어하지 못하는 검색 엔진 내의 검색어 데이터에 혼란을 주기 위함으로 이와 같은 환경을 설정하였다. 이렇게 생성된 계정에 기준 검색어 100개를 사용한 검색 활동을 진행하였다. 셀레니움의 웹 테스트 자동화 기능을 활용하여 검색 결과를 수집을 진행하였다. 기준 검색어의 첫 번째인 'Loan'을 구글 검색창에 입력하고 나온 결과 URL들의 정보가 포함된 HTML 문서 태그를 찾고 그 결과 값을 리스트 형식으로 가져와 Excel 내보내어 저장한다. 'Loan'의 결과 URL 100개가 저장되면 다음 검색어 'CEO'를 검색한다. 이 전과 같은 활동을 반복한다. 이렇게 한 검색어 당 100개의 URL을 저장한 Excel 데이터 100개가 만들어지고 Pure Account T1 Set으로 저장된다.

#### 3.2.3 Experimental Group

실험군의 프로세스는 그림 6과 같다.

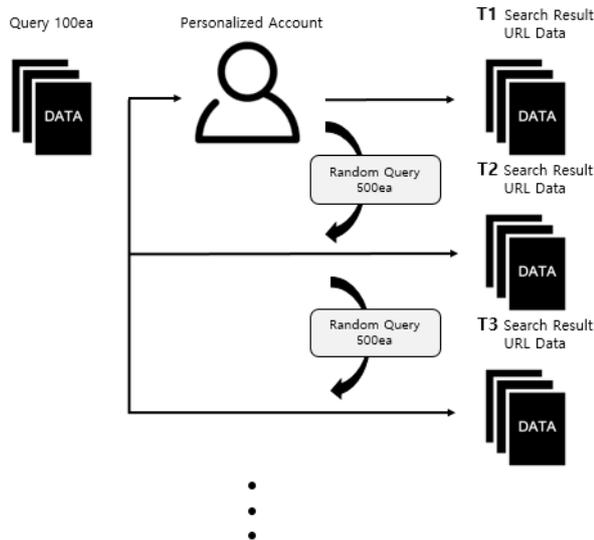


Fig. 6. Experimental Group of Control Group Search Activity

이 실험군은 이미 개인화가 이루어진 계정으로 검색 결과 URL 데이터를 수집하여 랜더마이징이 진행됨에 따라서 순위 변동을 분석한다. 이를 위해 하나의 계정을 사용하고 T1의 URL 수집이 완료되면 파이썬 라이브러리 내 RandomWords를 사용하여 단어들의 집합 중 하나의 단어를 무작위 추출하고 이를 검색창에 입력하여 500번의 검색 활동을 진행한다. 이후 URL을 수집하는 방식은 대조군에서 설명한 바와 같다. T가 증가하면서 서버 내에 축적이 되는 랜덤한 검색어의 수가 많아지고, 이에 따라 개인화 알고리즘의 혼란을 주기 위함이다.

### 3.3 Scoring

Excel 데이터로 저장된 URL들은 검색어에 따른 결과 URL과 URL 순위로 구성되어 있다. 이 데이터를 자바 프로그램 언어로 Import 하여 각각 List 배열로 저장한다. T1의 List 배열과 T2의 List 배열을 비교하여 같은 정보를 담고 있다면 T1의 순위와 T2의 순위의 차를 List 배열 내의 변수  $i$ 로 계산한다. 식 (1) STQScore(Static Query Score)는 각각의 기준 검색어에 대한  $T_n T_{n+1}$  사이의 순위 변동 값을 뜻한다. 각각의 기준 검색어 100개의 T1T2 사이의 변동 스코어를 합산하여 T1T2S를 추출하고 회를 거듭하면서 실험/대조군의 스코어의 변동을 분석하였다.

$$T_n T_{n+1} S = \sum_{n=1}^{100} STQ_n Score_{(1)}$$

각각의 T마다 수집된 URL 중 100순위 밖으로 밀려나는 데이터 또한 있을 것으로 보고 총 100개의 URL을 수집한 후 60개의 URL까지의 변동을 스코어링 하였다. 이때 상위에 랭크된 URL의 변화가 사용자에게 노출되는 빈도가 높고 영향력이 클 것이라 예상되어

$$R(x) \begin{cases} (T_n R - T_{n+1} R) * 0.3 & \text{when } 0 < x \leq 20 \\ (T_n R - T_{n+1} R) * 0.2 & \text{when } 21 < x \leq 40 \\ (T_n R - T_{n+1} R) * 0.1 & \text{when } 41 < x \leq 60 \end{cases} \quad (2)$$

식 (2)와 같이 60개의 URL을 3:2:1로 비례하여 순위별로 3분류로 나누어 가중치를 두었고, 이는 순위 변동의 수와 곱하여 점수가 산정되도록 하였다. 즉 1위와 20위 사이의 URL이 순위 변동이 일어난다면 순위의 차를 계산한 후 값에 0.3을 곱하여 높은 순위의 URL의 가중치를 두었다.

## IV. Experiment Result and Analysis

### 4. Experiment Result and Analysis

#### 4.1 Performance Evaluation Method

검색어 랜더마이징이 효과가 있다는 것을 증명하기 위해 실험/대조군의  $T_n T_{n+1}$  스코어를 비교한다. 각 T마다 산출되는  $T_n T_{n+1}$  스코어를 활용하여, 그래프로 표현하고, 실험군 랜더마이징 후의 스코어의 변화를 분석하였다. 스코어가 높을수록 URL 순위의 변동이 많았다는 것을 의미한다. 따라서 대조군보다 실험군의  $T_n T_{n+1}$  스코어가 높아지면 검색어 랜더마이징의 효과가 유의미한 것으로 판단한다.

#### 4.2 Experiment Analysis

그림 7은 T1과 T2 사이의 검색어 각각에 대한 점수 분포이다. 검색어마다 URL 변동의 폭은 각각 다르나 기준이 되는 분야별 100개의 검색어를 매번 새로운 계정과 개인화된 계정에 검색 활동을 하고, 그 결과 URL 100개를 저장하였다. 따라서 두 계정에 대한 시간 T당 1,000개의 URL이 저장된다. T1에서의 점수 분포는 대조군의 평균이 503.6, 실험군은 469.13으로 이미 개인화가 이루어진 실험군의 URL 다양성이 대조군보다 낮은 것을 확인할 수 있다.

그림 8은 T1 - T30까지의 결과  $T_n T_{n+1}$  스코어의 변화를 나타낸다. 초기 실험에서 랜더마이징이 거듭됨에 따라 실험군의 스코어가 초기보다 높아진 것으로 확인 되었다.

그림 9는 T30 -T60 사이에서의  $T_n T_{n+1}$  스코어를 나타낸 그래프이다. 그림 8에서 실험/대조군의 스코어 차가 미비하였지만, 거듭된 검색어 축적으로 인하여 실험군의 URL 변동 스코어가 대조군보다 높아지는 것을 확인하였다. 그러나 T37T38에서의 스코어는 대조군이 더 높았었고, T44T45에서는 근사값을 가짐으로 검색어 랜더마이징의 효과가 일정하다고 보기에는 한계가 있었다.

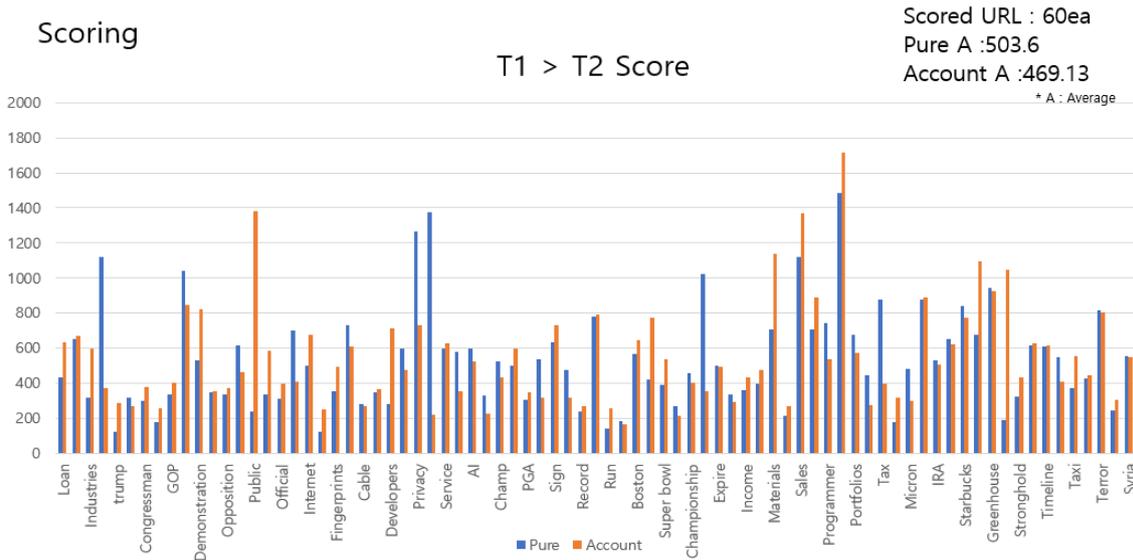


Fig. 7. URL Variation Score for The Static Query

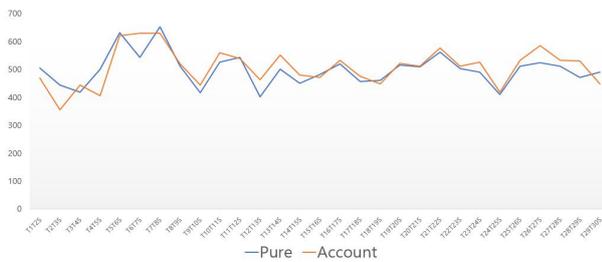


Fig. 8. TnTn+1S Graph ( T1 - T30 )

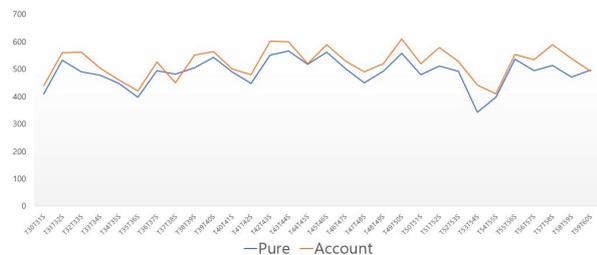


Fig. 9. TnTn+1S Graph ( T31 - T60 )

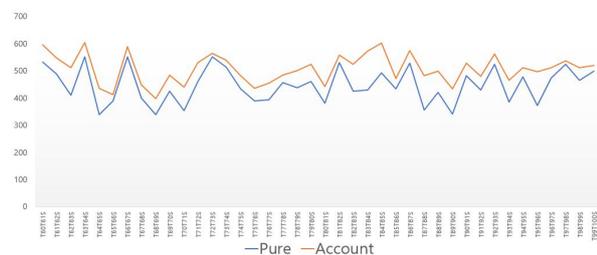


Fig. 10. TnTn+1S Graph ( T61 - T100 )

그림 10은 T60-T100까지의 결과 그래프이다. 60회를 진행하는 동안 축적된 랜덤한 검색어 수가 3만 개를 넘었고 이에 URL 스코어 또한 대조군보다 실험군의 스코어가 일정 수준 높게 유지되는 것을 확인하였다. 이후 T100까지의 이 스코어의 격차를 유지하였다. 이는 검색어에 대한 결과 URL의 다양성이

실험 초기보다 높아진 것을 의미한다. 다양성이 높아지므로, 사용자가 더 많은 URL에 노출되면서 검색 엔진의 개인화로 인해 정보의 폭이 좁아지는 것을 해소할 수 있었다.

### V. Conclusions

본 연구에서는 개인화 검색의 문제점에 대하여, 검색어를 랜더마이징하여 서버에 축적되는 검색어 데이터에 혼란을 주는 방법을 제안하고 실험하였다. 본 연구 방법은 이미 개인화되어 있는 계정도 효율적으로 적용될 수 있으며, 사용자가 인식하지 못한 상태에서, 개인 정보를 통하여 개인화를 시도하는 서비스 제공자에 대한 대비책을 마련하였다.

제안한 방법이 Random() 함수와 TimeSleep() 함수의 결합을 통하여 검색 자동화 테스트의 딜레이를 주는 방법으로 오토봇 인식문제(Captcha)를 보완하여 실험을 진행하였지만, 이를 완벽히 해결하지는 못하였다. 따라서 T를 100까지의 제한적인 실험을 진행하였고, 실험의 범위가 한정적이지만 검색어 랜더마이징이 개인화 알고리즘에 혼란을 주는 유의미한 결과를 도출하였다. T의 범위가 늘어날수록 축적되는 단어의 수가 증가하면서 랜더마이징의 효과가 커질 것으로 예측된다. 반면, 기존 검색어의 데이터가 T의 수만큼 쌓이게 됨으로, 기존 검색어의 관련된 개인화가 이루어질 가능성이 있다.

기대되는 응용 분야로는 사용자 모바일 폰 내에 서비스 형식의 어플리케이션을 구현하여 사용자가 모바일 폰을 사용하는 동안 연결된 계정의 검색어 랜더마이징을 가능케 하는 것을 기대할 수 있다. 이에 안드로이드 어플리케이션 상에서 사용자의 의식에 방해되지 않는 키 이벤트를 활용하여, 키를 누르는 동시에 랜덤 검색어가 서버로 전달되어 검색 활동을 하게 하는 어플의 개발을 향후 연구 목표로 하고 있다.

## REFERENCES

- [1] Pariser, E., "The filter bubble: What the Internet is hiding from you.", Penguin, UK, 2011.
- [2] Nagulendra, S., Vassileva, J., "Understanding and controlling the filter bubble through interactive visualization: A user study." In Proceedings of the 25th ACM conference on Hypertext and social media, pp.107-115, ACM. September 2014.
- [3] Xing, X., Meng, W., Doozan, D., Feamster, N., Lee, W., Snoeren, A. C., "Exposing inconsistent web search results with bobble." In International Conference on Passive and Active Network Measurement, pp.131-140, Springer, Cham, March, 2014.
- [4] Selenium Process, <http://www.seleniumhq.org/projects/remote-control/>
- [5] Colborn, K., "Guide to Personalized Search Results." Portent <https://www.portent.com/blog/seo/personalized-search-results.htm>.
- [6] Jeeyun O, Sunggwon P., "The Effects of Search Engine Credibility and Information Ranking on Search Behavior." Korean Journal of Journalism & Communication Studies, pp.26-49, 2009.
- [7] Labrou, Y., Finin, T., "Yahoo! as an ontology: using Yahoo! categories to describe documents." In Proceedings of the eighth international conference on Information and knowledge management, pp.180-187, Kansas City, Missouri, USA, November, 1999.
- [8] Horling, B., Kulick, M., "Personalized Search for everyone." The Official Google Blog, April, 2009.
- [9] Gottron, T., Schwagereit, F., "The Impact of the Filter Bubble---A Simulation Based Framework for Measuring Personalisation Macro Effects in Online Communities.", arXiv preprint, arXiv:1612.06551, 2016.
- [10] Wang, H., Shao, S., Zhou, X., Wan, C., Bouguettaya, A., "Preference recommendation for personalized search." Knowledge-Based Systems, Elsevier, pp.124-136, May 2016.
- [11] Junyeop, L., Joohong, L., YougSuk C., "Mitigating the filter bubble problem in recommendation system using natural language meta data." Communications of the Korean Institute of Information Scientists and Engineers, pp.766-768, November 2015.

## Authors



Sangdon Joo is an undergraduate student of Computer Science and Engineering at Gachon University, Korea.



Sukyung Seo is an undergraduate student of Computer Science and Engineering at Gachon University, Korea. Sukyung Seo is currently an undergraduate researcher in Data Mining & Bioinformatics laboratory, Gachon University. She is interested in

network biology and data mining.



Youngmi Yoon received the B.S. degree from Seoul National University in 1981; the M.S. degrees in statistics and computer science from Stanford University in 1984 and 1987 respectively, and the Ph.D. degree in computer science

from Yonsei University in 2008. Youngmi Yoon worked as a software engineer from 1987 to 1993 at IntelliGenetics Corp. in Mountain View, CA, USA. She's been a professor at Gachon University from 1995. Her research interest includes database, data science, data mining, and bioinformatics.