인명 검색 결과의 동적 클러스터링 방법

김형준O, 강승식
국민대학교 컴퓨터공학부
dictions@kookmin.ac.kr, sskang@kookmin.ac.kr

Dynamic Clustering Method for Person Name Search Result

Hyoung-Joon KimO, Seung-Shik Kang
School of Computer Science, Kookmin University

1. 서 론

웹(WWW)이 발전하면서 규모가 커지고 점점 세계인의 머릿속에는 세상의 모든 정보가 웹에 있다는 상식이 자리잡았다. 문제는 이 넓은 웹 속에 어느 곳에 어떤 정보가 있는가 하는 질문에서 시작해서 효율적인 검색을 통해 관심 키워드와 연관된 문서를 검색하더라도 그 결과의 양이 많아서 개인 사용자가 직접적으로 혹은, 실시간으로 수용할 수 없는 상황에 이르렀다.

본 논문은 사람의 이름을 검색한 결과에 대한 동적 클러스터링 시스템을 제안하고 구현한다. 이 시스템은 인명 검색 결과로 얻어지는 문서들에 출현하는 용어들 중 문서간의 공통 용어들을 바탕으로 그 유사한 정도를 추정함으로써 서로 관련된 문서들을 묶어낸다. 사용자는 관련 문서를 묶어서 브라우징함으로써 불필요한 문서는 한꺼번에 걸러내고 필요한 문서는 한꺼번에 찾는다. 결과적으로 효율적인 인터페이스를 통해 편리하고 빠른 검색이 가능하고 검색 서비스의 질이 개선된다.

2. 본론

검색서비스 제공자로부터 검색의 결과를 원문 전체와 함께 받아 사용하는 것은 오버헤드가 크다. 웹 검색의 결과를 처리하는 실시간 시스템에서 용어의 수는 처리속도를 저해하는 가장 큰 요인으로 꼽을 수 있다. 매일 수십억 개의 쿼리를 처리하며 수 천명이 동시에 서비스를 요청하는 야후와 같은 거대 검색 서비스의 입장에서 실시간 서비스의 처리속도는 아무리 강조해도 부족함이 없다 하겠다. 이런 점을 고려하여 본 연구에서는 검색 결과 문서로써 단문을 사용하는 것이 현실적이고 효율적이라고 판단한다.

특정 인명을 쿼리로하는 검색의 결과로 얻어진 단문 문서들로부터 색인어 추출기를 사용하여 용어를 추출하고 문서 별 용어 테이블(Term Table)로 관리하도록 하는 색인 과정이 가장먼저 이루어진다. 색인은 국민대학교 형태소 분석기와 한국어 분석 모듈(KLT 2.10b)의 색인어 추출기능을 사용하였다.[1] 이 과정에서 각 문서의 용어와 문서별 출현 빈도, 품사를 구하여 테이블로 구축하고, *DF*를 계산하여 용어 선택 과정 및 문서-문서, 클러스터-클러스터의 유사도 계산에 사용한다.

실시간으로 구축한 용어 테이블과 용어 사전을 통해 각 문서간의 공통 용어와 각 용어의 정보, 각 문서의 길이를 정리한다. 이 공통 용어들을 바탕으로 문서간의 유사도를 추정하고 특정 값 이상의 유사도를 갖는 문서들을 묶어서 각각의 클러스터로 구성한다.

문서간 공통용어의 가중치를 계산하기 위해 공통용어의 *tf*와 *df*를 사용하는 *TF-IDF* 가중치 계산 방법을 기본으로 용어의 품사에 따른 선택적 가중치를 적용하고 문서의 길이 및 공통 용어의 수, 공통용어의 품사를 반영하는 유사도 계산 함수를 사용한다.[2]

$$sim(d_x, d_y) = \frac{size(d_x \cap d_y)^2}{size(d_x) \times size(d_y)} \times \sum_{t_i = t_j \in d_x \cap d_y} \frac{tf_i}{df_i} \cdot p(pos_i) \times \frac{tf_j}{df_j} \cdot p(pos_j)$$

$$= \frac{size(d_x \cap d_y)^2}{size(d_x) \times size(d_y)} \times \sum_{t_i = t_j \in d_x \cap d_y} \frac{tf_i \cdot tf_j}{df^2} \cdot p^2 \qquad p(pos_i) = \begin{cases} 1.5 \; if \; pos_i = 'K' \\ 1.0 \; else \end{cases}$$

$d_x \cap d_y$: 문서 d_x, d_y의 공통 용어 집합
size(x) : 문서 혹은 용어 집합 x의 용어 수
pos_i : t_i의 품사

검색 결과의 문서집합을 포함한 대부분의 문서는 제목(title)과 본문(body)으로 구성되고, 제목은 문서 전체의

내용을 대표하는 경우가 많다. 이런 이유로 두 문서의 유사도 계산시 제목과 본문의 유사도를 각각 계산하고 제목의 유사도에 가중치를 주어 아래의 수식과 같이 수치화 하였다. 상수 k는 0과 1 사이의 값으로 $k = 0.5$이면 제목과 본문의 유사도 반영 비율은 같고 $k > 0.5$일때 제목에 가중치를 주게 된다. 본 연구에서는 $k = 0.7$을 사용하여 본문보다 제목의 유사도에 가중치를 두었다.

$$sim(d_x, d_y)_{doc} = k \cdot sim(d_x, d_y)_{title} + (1-k) \cdot sim(d_x, d_y)_{body}$$

기준문서를 임의로 정하고 나머지 문서들 중 임계치보다 큰 유사도를 갖는 대상 문서를 선택해 클러스터로 구축한다. 새로운 기준문서가 이미 하나의 클러스터에 포함된 경우 또 다른 임의의 문서를 기준문서로 정하고 기본연산을 반복한다. 이와 같은 클러스터링 방법은 일종의 단일패스 클러스터링 알고리즘으로써 클러스터의 크기를 방대하게 키우는 경향이 있다.[3] 그러나 본 논문에서는 정밀한 임계치 조절을 통해 클러스터의 크기를 최소화하고 클러스터의 수를 다수로 늘린 후 클러스터간 유사도 계산을 통한 클러스터 병합이라는 추가연산을 통해 안정적인 성능을 낸다.

클러스터링의 단위 연산인 문서간 유사도 계산에서 서로 다른 클러스터의 문서들을 하나의 클러스터로 판별하는 리스크를 최소화하는 정책을 사용하는 경우 각 클러스터의 크기가 작고 전체 클러스터의 수가 너무 많은 문제가 발생한다. 우리는 이 문제를 작은 클러스터들을 하나의 문서처럼 간주하여 다시 클러스터간 유사도를 계산하고, 이 결과를 바탕으로 작은 클러스터들을 더 큰 클러스터로 병합하였다.

클러스터쌍의 모든 클러스터 유사도를 문서간 유사도 계산식을 사용해 계산하고 그 중 가장 유사한 클러스터쌍을 새로운 클러스터로 병합하되 이미 병합된 클러스터는 클러스터쌍 유사도가 높다고 하더라도 다시 병합하지 않음으로써 하나의 클러스터의 크기를 키우지 않고 전체 클러스터들의 크기 편차를 최소화하는 방법을 사용했다. 이 방법은 임계치나 클러스터의 수를 지정하지 않고도 적정수의 클러스터를 자동으로 생성할 수 있고, 클러스터의 수를 고정하거나 임계치 조절을 하고자 한다면 임계치나 클러스터의 수를 지정하고 병합되어 새로 구축된 클러스터들에 대해 다시 클러스터 병합 연산을 수행하는 것이 가능하다.

실험은 구글의 웹문서, 뉴스, 블로그 검색 서비스에서 인명 검색 결과 491개의 단문 문서를 수집하여 사용하였다. 문서 수집은 뉴스, 블로그의 검색 결과를 RSS feed로 받아 자동수집 하였고, 클러스터를 구분하는 수작업을 통해 구축되었다.

단순 클러스터링만으로는 비교적 높은 정확도를 보였으나 클러스터가 너무 산개하여 클러스터링 시스템으로서 가치가 없었다. 클러스터링된 클러스터들을 병합하여 클러스터를 재구축한 후의 정확도가 79.5%, 클러스터링된 문서의 비율 83%로 비교적 안정적인 성능을 보여주면서 클러스터 수는 15개로 적당한 수의 클러스터를 생성하여 병합 후 전체적인 클러스터링 성능이 나아졌음을 볼 수 있다.

3. 결론

본 논문에서는 인명 검색에 대한 결과 문서들을 연관 문서들끼리 동적으로 클러스터링 하기 위해 각 문서의 길이와 문서 사이의 공통 용어 수, *TF*, *DF*, 품사를 반영하여 문서간의 유사도를 계산하여 작은 여러 개의 클러스터로 클러스터링하고 다시 이 클러스터들을 병합하는 방법으로 좋은 성능의 실시간 클러스터링 시스템이 될 수 있음을 보였다.

클러스터링의 정확도는 유사도 계산의 정확도에 영향을 받음을 확인하였나, 유사도 계산만 정확하다고 좋은 클러스터링 시스템이 되는 것이 아니라는 사실도 확인했다. 또, 검색 결과에 대한 클러스터링을 실시간으로 처리할 수 있도록 용어수가 적은 단문을 사용하고 용어 선별을 통해 클러스터링에 꼭 필요한 용어만 처리하도록 하여 처리 속도 최적화에 집중했다. 그 결과 일반적인 검색 엔진의 검색 결과에서 처음 10페이지 약 100문서에 대해서 비교적 좋은 성능의 클러스터링을 실시간으로 처리할 수 있었다. 그러나 너무 적은 문서당 용어 수로 인해 정확도 향상에 한계성을 인식하고 향후 과제로 동의어와 같은 언어자원을 활용한 적은 용어 수에도 정확도를 개선할 수 있는 방법의 연구가 필요하다.

참고문헌

[1] 강승식, "한국어 형태소 분석과 정보 검색", 홍릉과학출판사, 2002.

[2] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization", Proceeding of ICML-97, 14th International Conference on Machine Learning, Nashville, pp143-151, 1996.

[3] William B.Frakes, Ricardo Baeza-Yates, "Information Retrieval: Data Structures and Algorithms", Prentice-Hall, Inc. 1992.