단어간 연관성을 사용한 kNN 알고리즘

전승룡*, 이재문*, 오하령**
*한성대학교 멀티미디어공학과
**국민대학교 전자공학과
e-mail:jmlee@hansung.ac.kr

kNN Alogrithm by Using Relationship with Words

Seong Ryong Jeun*, Jae Moon Lee*, Ha Ryoung Oh**
*Dept of Multimedia Engineering, Hansung University
**Dept of Electronics Engineering, Kookmin University

요 약

본 논문은 연관규칙탐사 기술에서 사용되는 빈발항목집합과 동일한 개념으로 문서분류의 문서에서 빈발단어집합을 정의하고, 이를 사용하여 문서분류 방법으로 잘 알려진 kNN에 적용하였다. 이를 위하여 하나의 문서는 여러 개의 문단으로 나뉘어졌으며 각 문단에 나타나는 단어들의 집합을 트랜잭션화하여 빈발단어집합을 찾을 수 있도록 하였다. 제안한 방법은 AI::Categorizer 프레임워크에서 구현되었으며 로이타21578 데이터를 사용하여 학습문서의 크기에 따라 그 정확도가 측정되었다. 정확도의 측정된 결과로 부터 제안된 방법이 기존의 방법에 비하여 정확도를 개선한다는 사실을 알 수 있었다.

1. 서론

최근 웹 문서 등 전자 문서 급증과 인터넷을 통한 이들에 대한 정보 검색의 급증으로 이들을 관리하는 정보관리시스템 분야에서 문서 분류 연구가 활발히 진행되고 있다[1-6]. 문서 분류에 대한 연구의 방향은 크게 두 가지로 나뉜다. 하나는 분류의 정확도를 높이는 기술에 관한 연구이고[1, 6], 다른 하나는 문서 분류의 속도를 높이는 기술에 관한 연구이다[2, 3, 5]. 문서 분류에 대한 대부분의 연구는 전자에 집중되어 왔다. kNN[1, 5]은 가장 간단한 기법 중의 하나이면서 비교적 높은 분류 정확도를 보이지만, 실행 속도가 매우 느리다.

대부분의 기존 문서분류에서는 단어별 가중치만 적용하는 기법을 사용하고, 단어간 가중치를 적용하는 기법은 사용하지 않는다[1]. 이것의 이유는 문서에서 단어간의 가중치를 계산하는 것이 쉽지 않기 때문이다. 본 논문에서는 연관규칙탐사 기술에서 사용되는 빈발항목집합과 동일한 개념으로 문서분류의 문서에서 빈발단어집합을 정의하고, 이를 사용하여 단어간 가중치를 계산하여 kNN의 정확도를 높이고자 하는 것이다. 이를 위하여 연관규칙탐사 등 데이터마이닝에서 사용하는 빈발항목집합(Large Item Sets)[6, 7]과 동일한 개념의 빈발단어집합을 정의하고 이를 이용하여 단어간 가중치를 계산한다. 이를 위하여 문서에서의 빈발단어집합을 정의하고, 연관규칙탐사에서의 빈발항목집합을 찾는 방법을 이용하여 빈발단어집합을 찾는 방법을 제안하였다. 제안된 알고리즘은 잘 알려진 문서분류 프레임워크상에서 구현되었으며, 로이터-21578 데이터를 사용하여 실험되었다. 실험을 통하여 제안된 알고리즘이 기존의 kNN보다 정확도를 개선한다는 사실을 알 수 있었다.

2장에서는 kNN에 대한 전반적인 소개와 알고리즘을 소개하며 또한 빈발단어집합을 소개한다. 3장에서는 문서에서의 빈발단어집합을 정의하고, 빈발단어집합을 찾는 알고리즘을 소개한다. 4장에서는 기존의 kNN와 제안된 kNN 사이의 성능 비교를 하며, 5장에서 결론을 논한다.

2. 기존 연구

kNN은 학습 단계에서는 학습 문서에 대한 벡터화정도만 한다. 벡터화는 하나의 문서를 문서에 나타나는 단어들에 대하여 <단어, 가중치>의 목록으로 변환하는 것이다. 대부분 가중치는 해당 문서에 나타나는 발생수를 사용한다. kNN이 학습단계에서 학습문서에 대하여 벡터화 정도만 하므로 kNN이 어떻게 동작하는 가는 분류 단계만의 설명으로 충분하다. kNN은 분류 집합, 학습 문서 집합, 시험 문서 및 상수 k를 입력받는다. 다음은 [1, 5]에서 제시한 내용을 기초로 한 kNN에 대한 기술이다.

kNN(입력:분류C,학습문서D,시험문서d,이웃상수k,출력:등급화C)

스텝01 : d=*build_vector_normalize*(d)

스텝02 : D_k={};

스텝03 : (d_m, m_s)= (ϕ, 0); // m_s: 최소의 유사성

스텝04 : foreach $d_i \in D$ {

스텝05 : s_i= *compute_similarity*(d, d_i)

스텝06 : if($m_s < s_i$){

스텝07 : $D_k = D_k \cup \{<d_i, s_i>\} - \{d_m, m_s\}$
스텝08 : $(d_m, m_s) = find\_minimal\_similarity(D_k)$
스텝09 : }
스텝10 : }
스텝11 : *C의 계수를 0으로 초기화 한다.*
스텝12 : foreach $d_{si} \in D_k$ {
스텝13 : d_{si}*가 속한 모든* $c_i = \{c_{i1}, c_{i2}, \ldots, c_{in} \in C\}$*의 계수를 증가 시킨다.*
스텝14 : }
스텝15 : *C를 계수값으로 정렬하여 출력한다.*

상기 알고리즘에서 스텝01의 벡터화 및 정규화는 주어진 문서에 대하여 <단어, 가중치>를 찾고, 이의 크기를 1로 하는 것이다[1, 2, 3]. m_s는 D_k에서 가장 작은 유사성을 의미한다. 스텝05에서 *compute_similarity(d, d_i)* 함수는 두 입력 문서 *d*, d_i의 유사성(s_i)을 계산하는 함수이다. 스텝07에서 D_k는 *k*개의 가장 큰 유사성을 가지는 문서의 집합이고 *find_minimal_similarity*(D_k)는 D_k에서 가장 작은 유사성을 가지는 문서와 유사성을 찾는 함수이다. kNN에서 유사성의 계산으로 두 문서에 대한 벡터의 내적으로 한다[1, 2, 3].

2.2 빈발항목집합

빈발항목집합은 연관 규칙 탐사 문제에서 정의된 것이다[6, 7]. $I=\{i_1, i_2, \ldots, i_m\}$를 *m*개의 중복이 없는 항목의 집합이라 하자. 트랜잭션 *T*는 고유한 식별자를 가지고 있으며 *I*의 부분집합인 항목들의 집합이다. 데이터베이스 *D*는 이러한 트랜잭션 *T*의 집합이라고 하고, $|D|$를 *D*에 포함된 트랜잭션들의 개수라 할 때 $X \subseteq I$인 *X*가 $\sup(X) \geq |D| \times s$를 만족하면 항목집합 *X*를 ***빈발항목집합***이라 말한다[7]. 여기서 $\sup(X)$는 *D*의 트랜잭션 중 *X*를 포함하는 트랜잭션들의 개수를 의미하고, *s(%)*는 사용자가 지정하는 최소지지도이다.

3. kNN에 빈발단어집합의 적용

대부분의 기존 문서분류에서는 단어별 가중치만 적용하는 기법을 사용하고, 단어간 가중치를 적용하는 기법은 사용하지 않는다[1]. 이것의 이유는 문서에서 단어간의 가중치를 계산하는 것이 쉽지 않기 때문이다. 본 논문에서는 연관규칙탐사 기술에서 사용되는 빈발항목집합과 동일한 개념으로 문서분류의 문서에서 빈발단어집합을 정의하고, 이를 사용하여 단어간 가중치를 계산하여 kNN의 정확도를 개선하는 기법을 제안하고자 한다.

3.1 문서에서의 트랜잭션 정의

문서에 나타나는 빈발단어집합을 정의하고 탐사하기 위해서는 단어집합(*I*), 트랜잭션(*T*), 데이터베이스 (*D*)를 정의하여야 한다. 문서는 여러 개의 문단으로 구성되어 있고, 하나의 문단은 한개 이상의 문장으로 구성되어 있다. 또한 하나의 문장은 한개 이상의 단어로 구성되어 있다. 문서에서의 문단, 문장, 단어의 계층적 관계는 문서 작성에 대한 기본적인 규칙이다. 본 논문에서도 하나의 문단을 하나의 트랜잭션(*T*)으로 고려한다. 이 경우 하나의 문단에 속하는 모든 단어들은 그 트랜잭션에 속하는 단어들(*I*)이 된다. 또한 단어집합(*I*)는 모든 문서에 나타나는 단어들의 전체 집합이 되며, 데이터베이스(*D*)는 문단으로 구성된 문서가 된다.

제목: 부끄러운 세계 1위, 온실가스 증가 -- 조선일보 2007년9월10일 사설
亞太아태경제협력체(APEC) 정상들이 8일 시드니 APEC 첫날 회의에서 溫暖化온난화를 막기 위해 에너지 효율을 2030년까지 25% 올리고 山林산림을 2020년까지 2000만ha 늘리자는 '시드니선언'을 채택하기로 했다. APEC 21개국엔 온실가스인 이산화탄소 배출량 순위로 1위(미국) 2위(중국) 4위(일본) 10위(한국) 국가가 들어 있고 회원국 배출량을 다 합치면 세계 배출량의 60% 이상을 차지한다.
앞서 지난 3월 EU 정상회담은 온실가스 배출량을 2020년까지 ...(중략)
1997년 체결된 교토議定書의정서는 선진 38개국에만 2012년까지 온실가스 ... (중략)
EU는 기업 간 온실가스 배출權권 거래제를 도입했고 영국은 개인별 배출량 ... (중략)

(그림 1) 예제문서

그림 1의 예제문서는 2007년 9월 10일자 조선일보 사설 중의 하나이다. 이 문서는 4개의 문단으로 구성되어 있다. 따라서 앞의 정의에 따라 4개의 트랜잭션으로 구성된다고 할 수 있고, 각 트랜잭션에 포함된 단어들은 고유 명사와 동사들을 제외하는 특정 규칙을 적용하였다고 할 때 표 5와 같이 찾을 수 있다. 표 1에서 가중치는 단어의 발생수로 하였다.

<표 1> 예제 문서에 대한 데이터베이스

트랜잭션	<단어, 가중치> 목록
T_1	<국가, 1><미국, 1><에너지, 1><온난화, 1><온실가스, 1><이산화, 1><일본, 1><정상, 1><중국, 1><한국, 1><협력, 1><시드니, 2><배출, 3>
T_2	<미국, 1><이산화, 1><한국, 1><배출, 2><에너지, 2><온실가스, 2><일본, 2><EU, 2><정상, 3>
T_3	<국가, 1><규제, 1><선진, 1><온실가스, 1><체결, 1><협약, 1><교토의정서, 2><배출, 2><한국, 2><감축, 3><의무, 3>
T_4	<경제, 1><국가, 1><영국, 1><이산화, 1><일본, 1><정부, 1><캠페인, 1><EU, 1><국제, 2><배출, 2><온실가스, 2>

3.2 빈발단어집합을 찾는 알고리즘

2장에서 설명한 kNN 알고리즘의 가장 기본적인 데이터는 *compute_similarity(d, d_i)*를 계산하기 위하여 사용되는 문서의 <단어, 가중치> 목록이다. 따라서 본 논문에서는 kNN 알고리즘에서 문서를 벡터화할 때 <*빈발단어집합, 가중치*> 목록을 찾는 방법만 제시하기로 한다.

Find_LargeWordSet(입력 : 문서d,최소지지도s 출력 : 빈발단어집합 L)

스텝01 : 하나의 문서를 분석하여 문단별(TID) <단어, 발생수>의 목록을 만든다.

스텝02 : 문단별 <단어, 발생수>를 단어별 <TID, 발생수>의 목록으로 변환하고, 이들중 목록의 요소수가 $s^*|T|$보다 큰 경우 집합 L_1에 포함한다.

스텝03 : for(k=2; $|L_{k-1}|$!= 0; k++){

스텝04 : forall $l_1 \in L_{k-1}$ {

스텝05 : L_x={ $c \mid c \in L_{k-1}, c[1]= l_1[1]\text{^...^}c[k\text{-}2] = l_1[k\text{-}2]\text{^}c[k\text{-}1] \neq l_1[k\text{-}1]$}

스텝06 : forall l_2 in L_x {

스텝07 : $c= l_1[1]\ l_1[2]\text{...}\ l_1[k\text{-}1]\ l_2[k\text{-}1]$로 구성된 c를 생성한다.

스텝08 : c의 k-1 서브 단어집합들이 L_{k-1}에 모두 없으면 스텝06을 수행한다.

스텝09 : l_1의 TID목록과 l_2의 TID목록을 교집합하여 c의 TID목록을 생성한다.

스텝10 : c의 TID목록의 개수가 $s^*|T|$보다 크면 $L_k= L_k \cup \{c\}$를 한다.

스텝11 : }

스텝12 : }

스텝13 : }

스텝14 : L={ L_1, L_2, ..., L_k}을 출력한다.

빈발단어집합을 찾는 방법으로 [7]에서 제시한 *Partition* 방법을 사용한다. 이 방법은 비교적 구현이 쉽고, 모든 데이터베이스를 메모리에 상주하여야 하므로 비교젓 소규모 데이터베이스에 적합하다. *Find_LargeWordSet*은 [7]에서 제시한 *Partition* 방법을 변형한 빈발단어집합을 찾는 알고리즘이다. 예를 들어 표 1의 데이터베이스에서 최소지지도(s) 75%을 만족하는 단어에 대하여 이를 단어별 <TID, 가중치>로 변환하면 표 2를 얻을 수 있다.

<표 2> 단어별 <TID, 가중치> 목록, L_1

단어	<TID, 가중치> 목록
국가	$<T_1, 1><T_3, 1><T_4, 1>$
배출	$<T_1, 3><T_2, 2><T_4, 2>$
온실가스	$<T_1, 1><T_2, 2><T_3, 1><T_4, 2>$
이산화	$<T_1, 1><T_2, 1><T_4, 1>$
일본	$<T_1, 1><T_2, 2><T_4, 1>$
한국	$<T_1, 1><T_2, 1><T_3, 2>$

표 2에서 '국가'의 <TID, 가중치> 목록인 '$<T_1, 1><T_3, 1><T_4, 1>$'의 의미는 '국가'라는 단어는 T_1, T_3, T_4 문단에서 발생하였으며, 각각은 1, 1, 1번씩 나타났다는 것을 의미하고 있다. $s^*|T|$가 3(=4*75%)이고, '국가', '배출', '온실가스', 이산화', '일본', '한국'은 각각 3, 4, 4, 3, 3, 3번 다른 문단에서 나타나기 때문에 L_1={'국가', '배출', '온실가스', 이산화', '일본', '한국'}이 된다. 스텝05에서 $c[i]$, $l_1[i]$는 각각 c, l_1의 단어집합 중 i번째 단어를 의미한다. 표 2의 목록을 이용하여 '국가:배출'의 TID 목록을 얻기 위해서는 '국가', '배출'의 TID 목록인 {$<T_1, 1><T_3, 1><T_4, 1>$}와 {$<T_1, 3><T_2, 2><T_3, 1><T_4, 2>$}을 TID 관점에서 교집합하여 {$<T_1, 4><T_3, 2><T_4, 3>$}이라는 '국가:배출'의 TID 목록을 얻게 된다. 이 경우 {$<T_1, 4><T_3, 2><T_4, 3>$}는 $s^*|T|$를 만족하므로 L_2에 속하게 된다. 여기서 가중치는 합하는 것으로 하였다. 반면, '국가:이산화'의 TID 목록을 같은 방법으로 계산하면 {$<T_1, 2><T_4, 2>$}를 얻게 되고 이는 $s^*|T|$를 만족하지 못하므로 L_2에 속하지 못하게 된다. 표 3은 표 2에 대하여 *Find_LargeWordSet*을 이용하여 구해진 단어집합별 <TID, 가중치> 목록이다. 본 논문에서는 문서에 대한 벡터로써 L_1뿐만 아니라 L_2, L_3, .., L_x를 사용하자는 것이다.

<표 3> 단어집합별 <TID, 가중치> 목록

L_x	단어집합	<TID, 가중치> 목록
L_2	국가:온실가스	$<T_1,2><T_3,2><T_4,3>$
	배출:온실가스	$<T_1,4><T_2,4><T_4,4>$
	배출:이산화	$<T_1,4><T_2,3><T_4,3>$
	배출:일본	$<T_1,4><T_2,4><T_4,3>$
	온실가스:이산화	$<T_1,2><T_2,3><T_4,3>$
	온실가스:일본	$<T_1,2><T_2,4><T_4,3>$
	온실가스:한국	$<T_1,2><T_2,3><T_3,3>$
	이산화:일본	$<T_1,2><T_2,3><T_4,2>$
L_3	배출:온실가스:이산화	$<T_1,8><T_2,7><T_4,7>$
	배출:온실가스:일본	$<T_1,8><T_2,8><T_4,7>$
	배출:이산화:일본	$<T_1,8><T_2,7><T_4,6>$
	온실가스:이산화:일본	$<T_1,4><T_2,7><T_4,6>$
L_4	배출:온실가스:이산화:일본	$<T_1,16><T_2,15><T_4,14>$

4. 실험 및 성능비교

4.1 실험 환경 및 측정 요소

제안된 알고리즘과 기존의 알고리즘을 문서분류의 실험 데이터로 잘 알려진 로이터-21578 데이터를 사용하여 실험을 통하여 성능을 비교 한다[1, 4]. 이를 위하여 제안된 알고리즘을 [2]에서 개발한 AI::Categorizer 프레임워크를 사용하여 구현하였다. 실험에 사용한 데이터는 로이터-21578 ApteMod 버전[4]이다. 총 10,788개의 문서로 구성된 로이터-21578 데이터는 중 시험 문서로 788개를 선택하고, 나머지 10,000개의 문서를 임의적으로 선택하여 학습문서(DxK) 생성하였다. 여기서 DxK란 x천개의 문서로 학습문서가 구성되었다는 것을 의미한다.

문서 분류에서 정확도의 측정은 [1]에서 설명한 전통적인 방법을 사용하였다. 즉, 리콜(R:Recall), 정밀도(P:Precision)를 각 분류별로 측정하여 평균을 구하는 매크로 방법와 전체에 대하여 측정하는 마이크로 방법을 사용하였다. 리콜(R)과 정밀도(P)에 대하

여 F_1을 아래 식과 같이 계산하였다[1].

$$F_1 = \frac{2PR}{P+R}$$

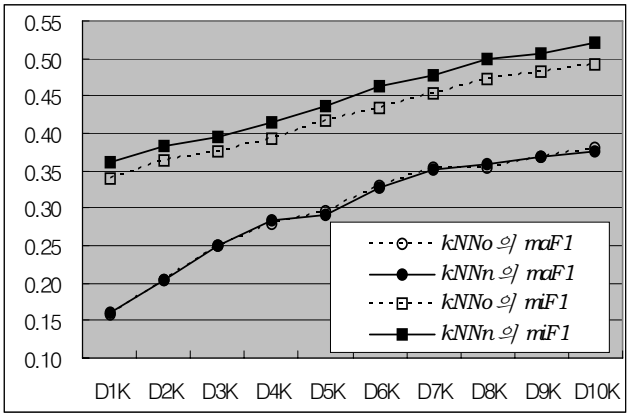
본 논문에서는 마이크로 F_1, 매크로 F_1만 측정하였으며, 각각 이들을 miF_1, maF_1로 표기한다. 기존의 kNN을 kNN_o로 표시하며 제안한 알고리즘을 kNN_p로 표시한다.

4.2 성능 측정

제안된 알고리즘에서 정확도는 최소지지도에 의존한다. 이것은 최소 지지도에 때라 생성되는 빈발단어집합의 수가 달라지기 때문이다. 빈발단어집합의 수가 크다는 것은 해당 문서의 특성을 나타내는 단어가 다양해진다는 것을 의미하며, 따라서 kNN의 정확도에 영향을 줄 수 있다. 표 4는 D9K에 대하여 최소 지지도를 10%에서 90%까지 변화 시키면서 maF_1 및 miF_1을 측정한 것이다. maF_1, miF_1 모두 최소지지도가 70%, 90%에서 보다 10%, 30%, 50%에서 더 좋은 성능을 보이는 것을 알 수 있다. 이것으로부터 빈발단어집합이 문서분류의 정확도 개선에 영향을 미친다는 사실을 알 수 있고, 또한 최소지지도가 감소할수록 정확도가 개선됨을 볼 수 있었다. 하지만 최소지지도의 감소는 빈발단어집합의 수를 증가시키므로 kNN의 속도를 감소시킨다. 따라서 kNN의 속도와 정확도의 개선에는 trade-off가 있다.

<표 4> 최소지지도 변화에 따른 maF_1, miF_1

학습데이터	최소지지도(%)	maF_1	miF_1
D9K	10	0.3457	0.4674
	30	0.3457	0.4674
	50	0.3457	0.4674
	70	0.3486	0.4663
	90	0.3486	0.4663



(그림 2) 학습문서의 크기(DnK)에 따른 F_1

다음 실험은 표 4로부터 최소지지도를 50%로 고정한 후 학습문서를 변경하면서 제안된 방법의 정확도를 측정한 것이다. 학습문서의 크기에 따른 F_1의 변화는 그림 2에서 보이고 있다. 그림 2에서 x축은 학습 문서의 크기를 나타내며, y축은 F_1의 값을 나타낸다. 그림 2를 통하여 알 수 있듯이, 매크로의 경우 kNN_o와 $kNNp$가 거의 비슷한 정확도를 나타내고 있다. 그러나 마이크로의 경우 확실히 $kNNp$가 kNN_o보다 높은 정확도를 준다는 것을 알 수 있다. 특히, 제안된 방법의 경우 학습문서의 크기가 클 때 더 좋은 정확도 개선효과를 준다는 것을 알 수 있다. 이것은 대부분의 문서분류에서 과도 학습을 나타내지 않는 학습문서의 범위 내에서는 학습문서가 많은 경우 더 좋은 정확도를 주기 때문이다.

5. 결론

본 논문은 연관규칙탐사 기술에서 사용되는 빈발항목집합을 변형하여 문서분류의 문서에서 빈발단어집합을 정의하였고, 이를 잘 알려진 kNN에 적용하여 이 방법의 정확도를 개선하였다. 이를 위하여 하나의 문서를 여러 개의 문단으로 나누었으며, 각 문단에 나타나는 단어들의 집합을 트랜잭션화하여 빈발단어집합을 찾을 수 있도록 하였다. 제안한 방법은 기존의 잘 알려진 문서분류 프레임워크에서 구현되었다. 또한 로이터-21578 데이터를 사용하여 그 정확도가 측정되었다. 실험은 학습문서의 크기를 변화하면서 기존의 kNN와 제안한 방법의 정확도를 측정하였다. 실험으로부터 마이크로 측정에서는 두 방법이 거의 유사한 정확도를 나타내었고, 매크로 측정에서는 제안한 방법이 많은 정확도 개선을 준다는 것을 알 수 있었다.

참고문헌

[1] Sebastiani F., "*Machine learning in automated text categorization*," ACM Computing Surveys, 34(1), pp.1-47, 2002.

[2] Williams K. and R. A. Calvo, "*A Framework for Text Categorization*," 7th Australian Document Computing Symposium., Dec., 2002.

[3] 이재문, "휴리스틱을 이용한 kNN의 효율성 개선", 정보처리학회논문지B, 제10-B권 제6호, 2003.

[4] Reuters-21578 Document Collection, http://about.reuters.com/researchandstandards/corpus.

[5] J.M. Lee, R. Calvo, "Scalable document classification", Intelligent Data Analysis: An International Journal, Vol. 9, No. 4, pp 365-380, 2005.

[6] 이재문, "빈발단어집합을 이용한 NaiveBayes의 정확도 개선", 한국인터넷정보학회 논문지, 169-178, 2006

[7] A. Savasere, E. Omiecinski and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", Proceedings of the 21th International Conference on Very Large Databases, pages 432-444, 1995.