대표용어를 이용한 *k*NN 분류기의 처리속도 개선

Improving Time Efficiency of kNN Classifier Using Keywords

이재윤, 연세대학교 문헌정보학과 강사
유수현, 연세대학교 대학원 문헌정보학과

Jae-Yun Lee, Su-Hyeon Yoo
Department of Library and Information Science, Yonsei University

*k*NN 기법은 높은 자동분류 성능을 보여주지만 처리 속도가 느리다는 단점이 있다. 이를 극복하기 위해 입력문서의 대표용어 *w*개를 선정하고 이를 포함한 학습문서만으로 학습집단을 축소함으로써 자동분류 속도를 향상시키는 kw_*k*NN을 제안하였다. 실험 결과 대표용어를 5개 사용할 경우에는 *k*NN 대비 문서간 비교횟수를 평균 18.4%로 축소할 수 있었다. 그러면서도 성능저하를 최소화하여 매크로 평균 F1 척도면에서는 차이가 없고 마이크로 평균정확률 면에서는 약 1~2% 포인트 이내로 *k*NN 기법의 성능에 근접한 결과를 얻었다.

1 서 론

최근에는 정보통신의 급격한 발달로 인해 수많은 정보를 손쉽게 접할 수 있게 되었다. 그러나 수많은 정보 중에 이용자에게 적합한 정보를 제공하기 위해서는 효율적인 정보관리 및 검색이 필요하다. 문서 범주화는 문서의 내용을 바탕으로 미리 정의된 범주를 문서에 부여함으로써 문서를 자동분류하는 기법이다(Yang and Pedersen 1997).

문서 범주화를 위해 보편적으로 이용되고 있는 학습 알고리즘에는 결정트리(Decision Tree) 기법, 신경망(Neural Network) 알고리즘, 나이브 베이즈(Naive Bayesian) 분류기, 최근접 이웃 분류기(*k*NN: *k*-Nearest Neighbor), 지지벡터기계(SVM: Support Vector Machines) 등이 있다. 이들 중에서 *k*NN과 SVM이 가장 좋은 성능을 보이는 것으로 알려져 있다(Yang and Liu 1999).

특히 *k*NN은 대표적인 예제기반 범주화 기법으로서 40여년 동안 패턴인식 분야에서 활발하게 연구가 이루어져 왔으며, 문서 범주화 영역에도 많이 응용되어 왔다. *k*NN은 입력문서가 주어졌을 때, 학습문서 중에서 입력문서와의 유사도가 가장 높은 *k*개의 문서를 추출하고 그들을 사용하여 각 후보 범주의 순위를 매기는 단순한 방법으로 구현이 용이하다. 그러나 *k*NN은 각 입력문서에 대

해 모든 학습문서를 비교해야 하기 때문에 선형분리기나 결정 트리 기법에 비해 처리속도가 느리다는 단점을 가지고 있다(Manning and Schütze 1999).

*k*NN의 이런 단점을 극복하기 위해서 이 연구에서는 입력문서의 대표용어를 이용하여 학습집단을 축소하는 kw_*k*NN 기법을 제안하였다. 학습집단의 축소는 분류성능의 저하를 가져오기 마련인데, 제안된 kw_*k*NN 기법이 *k*NN 기법의 느린 속도를 개선시키면서 분류 성능의 저하를 얼마나 보완하는지를 자동분류 실험을 통해 검증해보았다.

2 실험 설계

2.1 kw_*k*NN 기법

이 연구에서 제안하는 kw_*k*NN은 학습집단 중 입력문서의 대표용어를 포함하고 있는 문서들만을 대상으로 *k*NN을 수행하는 것이다. 즉, 입력문서의 대표용어 *w*개를 선정하고, *w*개의 용어 중 한 개라도 포함하고 있는 학습문서만을 추출함으로써 학습집단을 축소하고자 하는 것이다.

입력문서와 유사도가 높은 상위 *k*개의 학습문서는 입력문서의 대표용어를 한 개 이상 포함하고 있을 가능성이 높다. 왜냐하면 입력문서와 학습문서 간의 유사도는 두 문서간 일치하는 색인어의 가중치와 비례하기 때문이다. 따라서 상위 *k*개에 포함될 가능성이 낮은 학습문서들을 미리 배제할 수 있게 된다.

2.2 실험문서집단 처리

이 연구에서는 연세대학교 문헌정보학과에서 구축한 국제 및 경제 분야의 신문기사 말뭉치인 KFCM-CL 1020의 일부를 사용하였다. 이는 국내 3개 신문의 4월, 7월, 10월 기사 중에서 각 달의 1일부터 4일까지의 기사 340건씩을 모두 모아 1,020건을 구축한 것으로, 1992년도판 〈전국언론사 기사자료 표준분류표〉를 이용한 십진분류번호가 부여되어 있다. 이 연구에서는 각 대분류 내의 기사가 가장 많은 100번(정치), 200번(경제), 300번(산업), 900번(국제)을 택해서 실험집단을 구성하였다. 문서집단이 신문기사임을 감안하여 각 범주 내 최근기사 순으로 20%(178건)를 선정하여 검증집단을 구성하고, 나머지 80%(718건)를 학습집단으로 하였다. 〈표 1〉은 이 연구에서 사용한 실험문서집단의 구성을 정리한 것이다.

〈표 1〉 실험문서집단의 구성

대분류	문서 수		
	전체	학습집단	검증집단
100 (범주1)	167	134	33
200 (범주2)	330	265	65
300 (범주3)	116	93	23
900 (범주4)	283	226	57
합 계	896	718	178

한국어 형태소 분석기 HAM을 이용하여 불용어를 제거한 결과 총 색인어 29,828개를 추출하였고, 문헌빈도(DF)를 기준으로 분류자질을 선정하였다. DF가 2 이하인 저빈도어 24,088개와, 총 학습집단의 수(718)

의 약 10%에 해당하는 DF 72 이상의 고빈도어 87개를 제거한 결과 남은 색인어는 5,653개로 원래의 약 18.95%로 축소되었다. 〈표 2〉는 실험문서집단의 자질 통계를 나타낸다.

〈표 2〉 실험문서집단의 자질 통계

	학습집단	검증집단	합계
자질의 총 출현빈도	88,657	20,761	109,418
자질의 총 수	29,828	10,407	40,235
문서당 자질 수의 평균	41.54	58.47	44.91

축소된 자질들을 벡터로 표현하기 위해 TFIDF 가중치를 적용하였다. IDF 공식은 다음을 사용하였다.

$$IDF = \log_2 \frac{N}{df}$$

2.3 범주 판정 및 성능 평가

범주 판정을 위해 입력문서(Dx)와 학습문서(Dj)의 유사도를 다음과 같은 코사인 유사계수 공식으로 계산하였다. txk, tjk는 Dx, Dj에 출현한 용어 k의 가중치이다.

$$cosine(D_x, D_j) = \frac{\sum t_{xk} \times t_{jk}}{\sqrt{\sum (t_{xk})^2 \times \sum (t_{jk})^2}}$$

코사인 유사계수를 이용하여 입력문서와의 유사도 순으로 k개의 학습문서를 추출하는데, 이 연구에서는 k를 2에서 40까지 2씩 증가시키면서 성능의 변화를 측정하였다.

범주 판정 방법으로는 대표적으로 사용되는 범주별 유사도의 합(ss) 방법과, 문서의 범주빈도에 순위정보를 이용하는 이영숙, 정영미(2000)의 s1 방법을 사용하여 비교하였다. 각각의 공식은 다음과 같다.

ss 방법:

$$(C_k \mid D_x) \approx \sum sim(D_x, D_j) \times P(C_k \mid D_j)$$

s1 방법:

$$(C_k \mid D_x) \approx \sum \frac{P(C_k \mid D_j)}{rank(sim(D_x, D_j))}$$

kw_kNN의 경우 kNN과 동일한 전처리 방법을 사용하였다. 각 입력문서의 색인어 중에서 TFIDF 가중치가 높은 순위대로 w개를 선정한 후, 학습집단에서 이 w개의 용어 중 하나라도 포함한 문서들을 검색하여 리스트를 작성하였다. 이 연구에서는 w가 7인 kw7_kNN과 5인 kw5_kNN을 실험하였다. 각 입력문서에 대해 작성된 리스트를 학습집단으로 한 후 일반적인 kNN 방법과 동일한 실험절차를 거쳤다. 따라서 kw_kNN에서는 학습집단이 입력문서마다 다르게 구성된다.

평가방법으로는 범주별 성능을 먼저 계산한 다음 평균을 구하는 매크로 평균 척도와, 각 문서의 분류 결과를 평균하는 마이크로 평균 척도를 사용하였다. 매크로 평균 척도는 범주별로 계산된 정확률과 재현율을 이용하는 척도로서, 하나의 값으로 나타낼 때에는 범주별로 복합척도인 F_1 척도

값을 구해서 평균한 매크로 평균 F_1 척도를 사용한다. 마이크로 평균 척도는 각 문서단위로 구하는 것이므로 정확률과 재현율, F_1이 같다. 각 척도값을 구하는 공식은 범주가 C개인 경우에 아래와 같다.

범주 i의 정확률 $P_i = \frac{\text{범주 } i\text{에 분류된 적합문서 수}}{\text{범주 } i\text{에 분류된 문서 수}}$

범주 i의 재현율 $R_i = \frac{\text{범주 } i\text{에 분류된 적합문서 수}}{\text{범주 } i\text{에 속한 문서 수}}$

매크로 평균정확률 $= \frac{1}{C}\sum_{i=1}^{C} P_i$

매크로 평균재현율 $= \frac{1}{C}\sum_{i=1}^{C} R_i$

매크로 평균 $F_1 = \frac{1}{C}\sum_{i=1}^{C} \frac{2 \times P_i \times R_i}{P_i + R_i}$

마이크로 평균정확률 $= \frac{\text{범주 판정이 올바른 문서 수}}{\text{전체 문서 수}}$

3 실험결과 분석

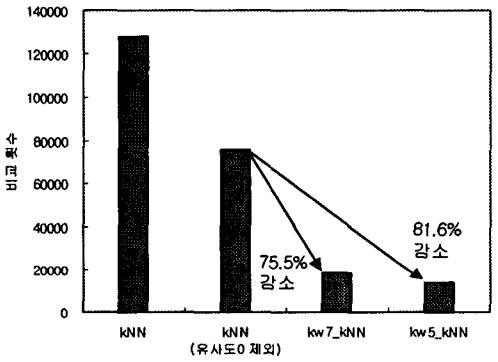
3.1 기법별 비교횟수

*k*NN과 kw7_*k*NN, kw5_*k*NN 기법을 사용할 때 각각 입력문서와 학습문서 사이의 비교횟수를 〈표 3〉에 제시하였다.

〈표 3〉 *k*NN 대비 kw_*k*NN의 비교횟수 비율

	*k*NN (127,804회 대비)	*k*NN(유사도0 제외) (75,458회 대비)
kw7_*k*NN (18,528회)	14.50%	24.55%
kw5_*k*NN (13,903회)	10.88%	18.42%

*k*NN의 경우, 분류를 위해 입력문서 당 학습문서의 유사도가 0 보다 큰 문서를 모두 비교해야 한다. 이 실험에서는 입력문서와의 유사도가 0인 학습문서를 제외할 경우에 *k*NN은 총 75,458회를 비교하였다. 반면에 kw_*k*NN은 입력문서 당 입력문서의 대표용어를 포함하는 학습문서에 대해서만 비교를 하므로 훨씬 적은 수만 비교하였다. 유사도가 0인 학습문서를 제외한 경우에 kw7_*k*NN은 *k*NN대비 24.55%, kw5_*k*NN은 *k*NN대비 18.42%만을 비교하는 것으로 나타났다(〈그림 1〉).



〈그림 1〉 *k*NN과 kw_*k*NN의 비교횟수 비교

3.2 성능 비교

대표용어의 수 w가 7인 경우(kw7_*k*NN)와 5인 경우(kw5_*k*NN)를 *k*NN과 함께 실험하되, 범주 판정을 유사도 순위를 이용하는 s1 방법에 의한 경우와 유사도를 이용하는 ss 방법에 의한 경우로 나누어서 살펴보았다.

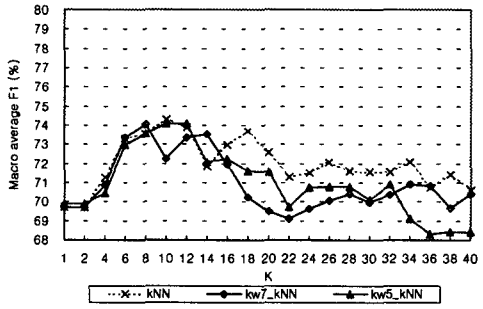
1) 범주 판정을 s1 방법으로 한 경우

*k*NN, kw7_*k*NN, kw5_*k*NN 기법을 적

〈표 4〉 기법별 분류 성능 - s_1 방법

척도	기법 \ k	1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
매크로 평균 정확률	kNN	68.97	68.97	70.69	73.26	73.79	75.27	75.77	74.25	74.45	76.52	75.81	73.99	75.51	77.41	76.93	76.94	76.94	77.42	76.60	76.95	76.59
	kw7_kNN	68.97	68.97	70.26	73.47	74.29	73.04	74.35	74.73	74.38	72.92	72.48	72.20	72.71	73.16	72.68	72.25	72.69	73.25	73.97	72.88	72.68
	kw5_kNN	69.13	69.13	69.86	73.11	73.79	74.14	74.76	73.52	74.31	74.81	75.06	72.66	73.70	74.78	74.78	73.12	73.59	72.12	71.62	71.64	71.64
매크로 평균 재현율	kNN	71.36	71.36	72.08	73.41	73.73	74.11	73.35	71.55	73.01	73.40	72.31	71.22	71.35	71.73	71.29	71.24	71.24	71.68	70.59	71.18	70.53
	kw7_kNN	71.36	71.36	71.70	73.41	74.16	72.20	73.23	73.34	71.55	70.08	69.43	68.94	69.38	69.76	70.20	69.76	70.15	70.59	70.53	69.34	70.20
	kw5_kNN	71.68	71.68	71.26	72.97	73.73	74.37	73.94	71.87	71.87	71.17	71.60	69.75	70.58	70.52	70.52	70.08	70.79	68.94	68.24	68.61	68.61
매크로 평균 F_1	kNN	69.71	69.71	71.27	73.31	73.55	74.28	73.87	71.84	72.98	73.70	72.57	71.28	71.51	72.09	71.58	71.54	71.54	72.06	70.73	71.43	70.61
	kw7_kNN	69.71	69.71	70.86	73.37	74.06	72.23	73.34	73.52	71.94	70.23	69.52	69.13	69.64	70.05	70.40	69.96	70.37	70.90	70.83	69.69	70.40
	kw5_kNN	69.90	69.90	70.45	72.97	73.55	74.09	74.07	72.06	72.25	71.60	71.57	69.72	70.74	70.78	70.78	70.10	70.91	69.09	68.30	68.42	68.42
마이크로 평균 정확률	kNN	72.47	72.47	74.16	76.40	76.40	76.97	76.97	75.84	76.40	76.97	76.40	75.84	76.97	77.53	76.97	76.97	76.97	77.53	76.97	76.97	76.97
	kw7_kNN	72.47	72.47	73.60	76.40	76.97	75.28	75.84	76.40	75.84	74.72	74.72	74.16	74.72	75.28	75.84	75.28	75.84	76.40	76.40	75.28	75.84
	kw5_kNN	72.47	72.47	73.03	75.84	76.40	76.40	75.84	75.28	75.28	75.28	75.84	74.72	75.84	75.84	75.84	75.28	75.28	74.16	74.16	74.16	74.16

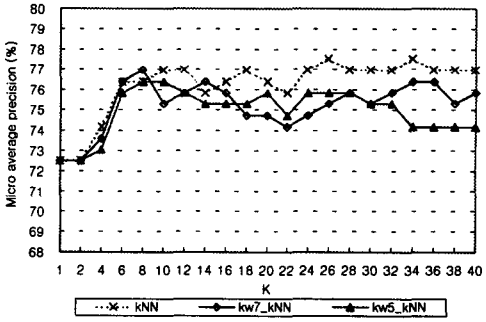
용하여 s1 방법으로 범주를 판정한 경우의 성능을 〈표 4〉에 제시하였다. 이중에서 매크로 평균 F_1 척도값을 〈그림 2〉에, 마이크로 평균정확률을 〈그림 3〉에 나타냈다.



〈그림 2〉 기법별 매크로 평균 F_1 - s_1 방법

s1 방법으로 범주를 판정한 경우, kw_*k*NN은 *k*가 16일 때까지는 *k*NN과 비슷한 성능을 보였다. 이 구간에서는 매크로 평균 F_1 척도 면에서 *k*NN과 kw5_*k*NN의 성능 차이가 없었으며, *k*가 1, 2, 12, 14일 때 *k*NN보다 더 높은 성능을 보였다. 마이크로 평균정확률 면에서는 kw5_*k*NN의 성능이 *k*NN에 비해 같거나 약 1%포인트 정도 낮았다.

*k*가 18 이상이 되면 *k*NN 대비 kw_*k*NN의 성능이 2내지 3%포인트 내외로 낮아진다. kw_*k*NN이 학습집단을 제한하는 기법이므로, *k*가 커질 경우에 유사도가 다소 낮은 문서까지 분류 단서로 사용하는 *k*NN에 비해서 성능이 낮아지는 것은 당연하다. 다만, 매크로 평균 F_1 척도와 마이크로 평균정확률을 둘 다 고려한다면 *k*를 큰 값으로 사용할 필요는 없을 것이다.

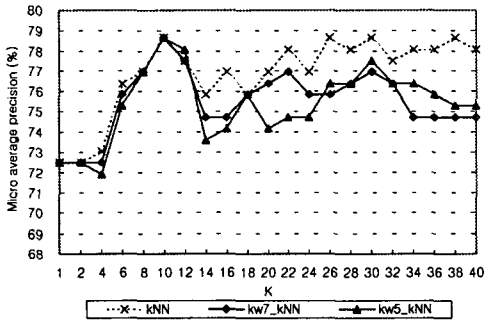


〈그림 3〉 기법별 마이크로 평균정확률 - s1 방법

〈표 5〉 기법별 분류 성능 - ss 방법Z

척도	기법 \ k	1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
매크로 평균 정확률	*k*NN	68.97	68.97	69.27	73.87	75.54	78.31	77.09	75.28	76.62	76.69	77.66	78.57	76.15	79.07	78.59	78.90	78.11	78.70	78.70	78.86	78.43
	kw7_*k*NN	68.97	68.97	68.81	73.78	75.41	77.67	76.56	72.78	72.91	73.74	75.11	74.96	73.40	73.66	74.14	74.50	74.17	72.52	73.35	72.76	72.92
	kw5_*k*NN	69.13	69.13	68.41	73.40	75.41	77.67	77.16	72.21	73.21	74.52	71.84	72.43	72.46	74.95	76.07	76.91	75.95	76.08	74.48	73.66	73.65
매크로 평균 재현율	*k*NN	71.36	71.36	70.61	73.67	74.38	75.38	74.12	71.51	73.34	71.45	72.96	73.78	72.32	74.17	73.41	73.85	72.97	73.41	73.41	73.47	72.71
	kw7_*k*NN	71.36	71.36	70.22	73.29	74.43	76.29	74.82	70.69	71.06	71.88	72.59	73.34	71.18	71.87	71.94	72.32	71.88	69.60	69.60	69.27	69.27
	kw5_*k*NN	71.68	71.68	69.78	72.85	74.43	76.29	75.91	70.57	70.68	71.88	70.99	71.75	70.35	71.88	72.20	73.03	71.83	71.83	71.45	70.35	70.35
매크로 평균 F_1	*k*NN	69.71	69.71	69.80	73.60	74.47	76.16	74.93	72.30	73.70	72.13	73.25	74.18	72.80	74.59	74.03	74.49	73.57	74.19	74.19	73.92	73.42
	kw7_*k*NN	69.71	69.71	69.39	73.22	74.46	76.49	75.35	71.14	71.35	72.17	72.95	73.38	71.50	71.86	72.23	72.54	72.13	70.66	70.21	69.47	69.58
	kw5_*k*NN	69.90	69.90	68.98	72.81	74.46	76.49	76.28	70.88	71.06	72.35	70.83	71.43	70.58	72.30	72.58	73.41	72.38	72.48	71.84	70.68	70.68
마이크로 평균 정확률	*k*NN	72.47	72.47	73.03	76.40	76.97	78.65	77.53	75.84	76.97	75.84	76.97	78.09	76.97	78.65	78.09	78.65	77.53	78.09	78.09	78.65	78.09
	kw7_*k*NN	72.47	72.47	72.47	75.84	76.97	78.65	77.53	74.72	74.72	75.84	76.40	76.97	75.84	75.84	76.40	76.97	76.40	74.72	74.72	74.72	74.72
	kw5_*k*NN	72.47	72.47	71.91	75.28	76.97	78.65	78.09	73.60	74.16	75.84	74.16	74.72	74.72	76.40	76.40	77.53	76.40	76.40	75.84	75.28	75.28

kw5_*k*NN과 kw7_*k*NN의 성능은 우열을 가리기가 어려웠다. *w*가 5인 경우가 7인 경우에 비해서 더 적은 학습문서를 활용하는데도 불구하고 성능이 저하되지 않은 것이다. 따라서 이 연구처럼 신문기사를 대상으로 할 경우에는 kw_*k*NN에서 *w*값은 5로 하더라도 충분하다고 생각된다.

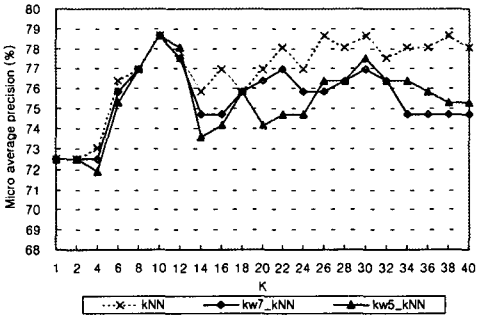


〈그림 4〉 기법별 매크로 평균 F1 - ss 방법

2) 범주 판정을 ss 방법으로 한 경우

*k*NN, kw7_*k*NN, kw5_*k*NN 기법을 적용하여 ss 방법으로 범주를 판정한 경우의 성능을 〈표 5〉에 제시하였다. 매크로 평균 F_1을 〈그림 4〉에, 마이크로 평균정확률을 〈그림 5〉에 나타내었다.

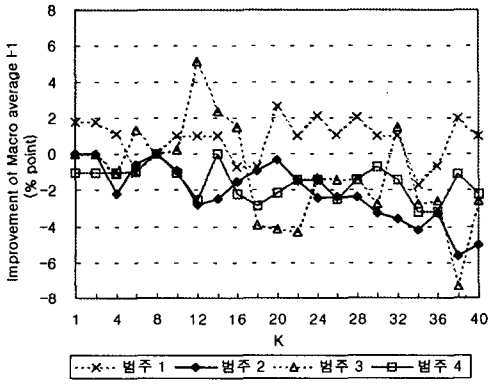
전체적인 성능은 *k*NN과 kw_*k*NN 모두 s1에 비해서 ss 방법으로 범주를 판정한 경우가 1~2%포인트 정도 높았지만, 일부 *k* 값의 경우에는 s1 방법의 결과가 더 좋기도 하였다.



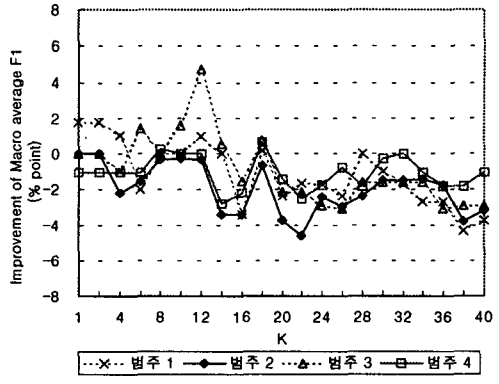
〈그림 5〉 기법별 마이크로 평균정확률 - ss 방법

s1 방법에서와 마찬가지로 ss 방법으로 범주를 판정한 경우에도, kw_*k*NN은 *k*가

작은 값일 때에는 kNN과 비슷한 성능을 보였다. 매크로 평균 F_1에 있어서 k가 1, 2, 10, 12일때 kw5_kNN이 더 높은 성능을 보였다. 마이크로 평균 정확률 면에서는 kw5_kNN의 성능이 kNN과 같거나 약간 낮았는데, k가 1, 2, 8, 10일 때 같았고 k가 12일 때에는 더 높았다.



〈그림 6〉 kNN 대비 kw5_kNN의 범주별 F_1 척도의 향상값(% point) - s1 방법



〈그림 7〉 kNN 대비 kw5_kNN의 범주별 F_1 척도의 향상값(% point) - ss 방법

특히 k가 작은 구간에서 kw5_kNN과 kNN의 마이크로 평균정확률을 비교했을 때, 범주 판정을 ss 방법으로 한 경우가 s1 방법을 사용한 경우보다 성능 차이가 적은 것으로 보인다.

k가 14 이상이 되면 ss 방법에서도 kNN 대비 kw_kNN의 성능이 2내지 3%포인트 이상 낮아진다.

3) 종합 비교

s1과 ss 방법 모두에서 kNN의 매크로와 마이크로 평균정확률이 공통적으로 좋은 지점은 k가 10일 때이다. 이때 kw5_kNN의 성능을 kNN의 성능과 비교해보면 s1 방법의 경우에 매크로 평균 F_1 척도에서 0.19%포인트 낮고, 마이크로 평균정확률에서 0.32%포인트 높다. ss 방법의 경우에는 kw5_kNN이 매크로 평균 F_1 척도에서 0.33%포인트 높고, 마이크로 평균정확률은 같다. 성능이 가장 좋은 k값에서는 kw_kNN의 성능이 kNN과 대등한 것이다.

4) 범주별 성능 분석

〈그림 6〉과 〈그림 7〉에서는 각 범주별로 kw5_kNN의 F_1 척도값에서 kNN의 F_1 척도값을 뺀 결과를 나타내었다. 여기에서 k가 14 이하인 경우에는 범주1과 3의 F_1값이 kw_kNN에서 더 높은 것으로 나타난다. 그런데 범주1과 3은 네 범주 중에서 크기가 작은 범주들이다. 즉, kw_kNN은 작은 범주에 대해서 성능 개선 효과를 보이는 것이다. 일반적으로 kNN기법은 학습문

서가 많이 속한 큰 범주의 성능이 높은 경향이 있는데, kw_*k*NN에서는 학습집단의 크기를 축소하기 때문에 이런 경향을 다소 완화시키는 것으로 짐작된다.

4 결 론

이 연구에서는 *k*NN 분류기의 자동분류 속도를 향상시키기 위한 방법으로 입력문서의 대표용어를 이용하는 kw_*k*NN을 제안하였다. kw_*k*NN은 입력문서의 대표용어 *w*개를 가중치가 높은 순서대로 선정하고 이 용어를 포함하고 있는 학습문서들만을 추출하여 *k*NN을 수행하는 것으로, 이 연구에서는 대표용어가 7개, 5개인 경우를 실험하였다.

실험 결과 대표용어를 이용한 kw_*k*NN은 *k*NN과 매크로 평균 F_1 척도 및 마이크로 평균정확률에 있어서 큰 차이가 없었으며, *k*가 작은 경우에 거의 동일한 성능을 보였다. 대표용어 5개를 이용한 실험은 7개를 이용한 실험과 별 차이가 없었다.

이와 같이 *k*NN에 근접한 성능을 보이면서도 입력문서와 학습문서간의 비교횟수에 있어서 대표용어 5개를 이용한 kw5_*k*NN은 *k*NN의 약 18.4%에 불과하였다.

앞으로 많은 문서를 포함한 다양한 실험집단에 대해 kw_*k*NN의 성능을 검증하는 실험이 필요할 것이다.

참 고 문 헌

이영숙, 정영미. 2000. "kNN 분류기의 범주할당 방법 비교 실험". 제7회 한국정보관리학회 학술대회 논문집, 37-40.

Manning, C.D., and H. Schütze. 1999. Foundations of Statistical Natural Language Processing, (Cambridge, Mass.: MIT Press).

Yang, Y., and X. Liu. 1999. "A re-examination of text categorization methods". Proceedings of the 22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 42-49.

Yang, Y., and J.O. Pedersen. 1997. "A comparative study on feature selection in text categorization". Proceedings of the Fourteenth International Conference on Machine Learning, 412-420.