문단 단위 가중치 함수와 문단 타입을 이용한 문서 범주화

주 원 균[†] · 김 진 숙[††] · 최 기 석[†††]

요 약

문서 범주화 분야에 대한 연구들은 전체 문서 단위에 한정되어 왔으나, 오늘날 대부분의 전문들이 주요 주제들을 표현하기 위해서 조직화된 특정 구조로 기술되고 있어, 텍스트 범주화에 대한 새로운 인식이 필요하게 되었다. 이러한 구조는 부주제(Sub-topic)의 텍스트 블록이나 문단(Passage) 단위의 나열로서 표현되는데, 이러한 구조 문서에 대한 부주제 구조를 반영하기 위해서 문단 단위(Passage-based) 문서 범주화 모델을 제안한다. 제안한 모델에서는 문서를 문단들로 분리하여 각각의 문단에 범주(Category)를 할당하고, 각 문단의 범주를 전체 문서의 범주로 병합하는 방법을 사용한다. 전형적인 문서 범주화와 비교할 때, 두 가지 부가적인 절차가 필요한데, 문단 분리와 문단 병합이 그것이다. 로이터(Reuter)의 4가지 하위 집합과 수십에서 수백 KB에 이르는 전문 테스트 컬렉션(KISTI-Theses)을 이용하여 실험하였는데, 다양한 문단 타입들의 효과와 범주 병합 과정에서의 문단 위치의 중요성에 초점을 맞추었다. 실험한 결과 산술적(Window) 문단이 모든 테스트 컬렉션에 대해서 가장 좋은 성능을 보였다. 또한 문단은 문서 안의 위치에 따라 주요 주제에 기여하는 바가 다른 것으로 나타났다.

키워드 : 문서 범주화, 문단, 비중첩 윈도우, 중첩 윈도우, 단락 검색, 경계 단락, 페이지, 텍스트 타일, 문단 가중치 함수

Automatic Text Categorization Using Passage-based Weight Function and Passage Type

WonKyun Joo[†] · JinSuk Kim[††] · KiSeok Choi[†††]

ABSTRACT

Researches in text categorization have been confined to whole-document-level classification, probably due to lacks of full-text test collections. However, full-length documents available today in large quantities pose renewed interests in text classification. A document is usually written in an organized structure to present its main topic(s). This structure can be expressed as a sequence of sub-topic text blocks, or passages. In order to reflect the sub-topic structure of a document, we propose a new passage-level or passage-based text categorization model, which segments a test document into several passages, assigns categories to each passage, and merges passage categories to document categories. Compared with traditional document-level categorization, two additional steps, passage splitting and category merging, are required in this model. By using four subsets of Reuters text categorization test collection and a full-text test collection of which documents are varying from tens of kilobytes to hundreds, we evaluated the proposed model, especially the effectiveness of various passage types and the importance of passage location in category merging. Our results show simple windows are best for all test collections tested in these experiments. We also found that passages have different degrees of contribution to main topic(s), depending on their location in the test document.

Key Words : Text Categorization, Passage, Non-overlapping Window, Overlapping Window, Paragraph, Passage Retrieval, Page, TextTile, Passage Weight Function

1. 서 론

문서에 하나 이상의 범주를 할당하는 문서 범주화(Text Categorization)는 정보검색과 기계 학습 분야에 있어서 중요한 연구 분야의 하나이다. 지금까지 텍스트 범주화의 연구 관심은 자질 추출(Feature Extraction), 자질 선택(Feature Selection), 지시된 학습 알고리즘(Supervised Learning Algorithm)과 하이퍼텍스트(Hypertext) 분류에 한정되어 왔고, 전형적인 범주화 시스템은 임의 문서 전체를 범주화의 한 단위로 사용하였다. 그러나 오늘날 많은 분야에서 평범하고 크기가 작은 문서 대신 다양한 포맷을 지니고 문서 크기가 수 MB에 이르는 워드 프로세스 문서, 전문 SGML/XML/HTML 문서, PDF/Postscript문서가 사용되고 있어서 전통적인 범주화 방법들에 대한 대안이 필요하게 되었다. 이러한 대안의 하나로서 각각의 문서를 연속적인 텍스트 조

† 정 회 원 : 한국과학기술정보연구원 연구원
†† 정 회 원 : 한국과학기술정보연구원 선임연구원
††† 정 회 원 : 한국과학기술정보연구원 국가RnD시스템개발실 실장
논문접수 : 2005년 3월 2일, 심사완료 : 2005년 7월 5일

각들인 문단(Passage)의 집합으로서 규정하는 접근 방법을 시도하게 되었다.

정보검색 분야에서 문단은 1990년 초기에 소개되었다[3, 4]. Wilkinson은 전문 검색에서 문서의 가중치 값을 결정하는 데 있어 문서의 구성 부분들의 점수를 조합하는 방법을 제안하였고[22], Callan은 다양한 종류의 문단에 대해 소개하였다[19]. 장문의 전문 문서 검색에서의 문단들의 영향력에 대한 많은 연구 결과가 수행되었다[20, 21]. 전문 검색 성능의 개선을 위해 다양한 문단 타입들을 사용하는 방법들이 시도되었다[3, 5, 6].

후에 문서 범주화(Txt Categorization) 관련 분야에서도 이와 비슷한 연구들이 수행되었다. Ludovic는 문서는 적합 단어 집합들과 부적합 단어 집합들로 구성됨을 전제로 하였고, 적합 단어 집합에 대해서 다른 가중치 접근 방법을 사용하여 문서 범주를 결정하는 방법을 시도하였다[23]. Chen은 문서를 관련 있는 단어들의 클러스터 그룹으로 구분하고 각 클러스터를 문서의 자질로 사용하여 분류하는 방법을 사용하였다[24]. Thanaruk은 장문의 웹 문서를 대상으로 텍스트 마이닝(Text Mining)을 수행하기 위해서 문서를 고정 길이 윈도우 방식의 문단으로 분리하였고, 문단 정보와 문단 간의 단어 상호 출현도(Co-occurrence)의 두 가지 정보를 이용하는 방식에 관해 연구하였다[25]. 이처럼 장문 문서들에 대한 연구는 정보 검색 전반에 걸쳐서 특히 텍스트 범주화 분야에서 새로운 관심을 끌고 있다.

본 논문에서는 단문 및 전문 모두에 효과적으로 사용할 수 있는 새로운 텍스트 범주화 모델을 제안한다. 제안하는 텍스트 범주화 모델은 문서를 문단으로 분리하고, 각 문단에 대해 범주 할당 작업을 수행하여 범주를 할당하고, 각 문단들의 범주를 한 문서의 범주로 병합하는 방법을 사용한다. 이러한 모델을 "문단 단위 문서 범주화(Passage-based Text Categorization)"라 정의하고, 문서 범주화에 기초한 전통적인 방법들과 범주화 성능을 비교하였다. 실험 결과에서는 kNN(K-Nearest Neighbor)[14] 문서 범주화 방법을 이용하여 다양한 문단 타입(Passage Types)과 문단 위치에 따른 텍스트 범주화 성능을 비교·분석 하였다. 문단을 사용함으로써 모든 문단 타입에 대해서 문서 단위의 범주화에 비해 만족할 만한 성능 향상을 보았다. 매우 큰 문서들의 경우에는 모든 문단 타입에 대해서 문서 단위의 범주화에 비해 10% 정도의 문서 범주화 성능 향상을 보았고, newswire와 같이 비교적 짧은 문서로 구성된 컬렉션(Collection)의 경우에도 5% 정도의 성능향상을 보았다.

2장에서는 문단 단위 문서 범주화 모델에 대해 소개하고, 3장에서는 문서 범주를 결정하는 방법에 대해서, 4장에서는 실험에 사용한 데이터 집합 및 성능평가방법에 대해 설명하고, 5장에서 결론을 맺는다.

2. 문단 단위 문서 범주화 모델

문서의 저자는 한 문서를 구성할 때, 고의적으로 여러 부주제들을 나열함으로써 하나 이상의 주요 주제어에 대해 설명한다[5]. 이렇게 구성된 문서는 임의 단위[1]의 부주제 블록의 연속적인 나열로서 설명될 수 있다. 이러한 동기에서 "문단 단위 문서 범주화 모델"을 제안하는데, (그림 1)에 묘사되어 있다.

정보 검색과 문서 범주화 모델 사이에는 문단을 이용하는 방식에서 기본적인 차이점이 발생하는데, 정보 검색에서는 데이터베이스에 저장 된 문서를 문단으로 분리하고 각 문단들에 대해서 질의 유사도를 계산한다[3, 4, 10, 18]. 반면 문서 범주화에서는 범주화 대상 문서를 문단으로 분리하고 전체 문서 대신 각 구성 문단을 대상으로 텍스트 범주화를 수행한다.

(그림 1)에서와 같이 문단 단위 문서 범주화 시스템은 하나의 대상 문서를 몇몇 문단으로 분리한 후 각각의 문단에 범주를 할당하고, 각각의 문단 범주를 병합함으로써 대상 문서에 대한 범주를 결정한다. 이러한 과정은 전형적인 문서 범주화 모델과 비교할 때, *문단 분리(Passage Splitting)*와 *범주 병합(Category Merging)*이라는 두 가지의 부가적인 절차가 수반된다. 두 절차는 각각 2.1절과 2.2절의 주제로서 설명한다.

2.1 문서 범주화에서 사용 하는 문단

(그림 1)의 문단 분리 측면에서 볼 때 문단이라는 개념은 문단 단위 문서 범주화의 핵심이라 볼 수 있다. Kaszkiel[7]은 문단을 한 문서 내의 연속적인 임의 텍스트 블록으로 정의하였고, Callan[3]은 문단 타입을 논리적(Discourse) 문단, 의미론적(Semantic) 문단, 산술적(Window) 문단의 세 종류로 구분하였다.

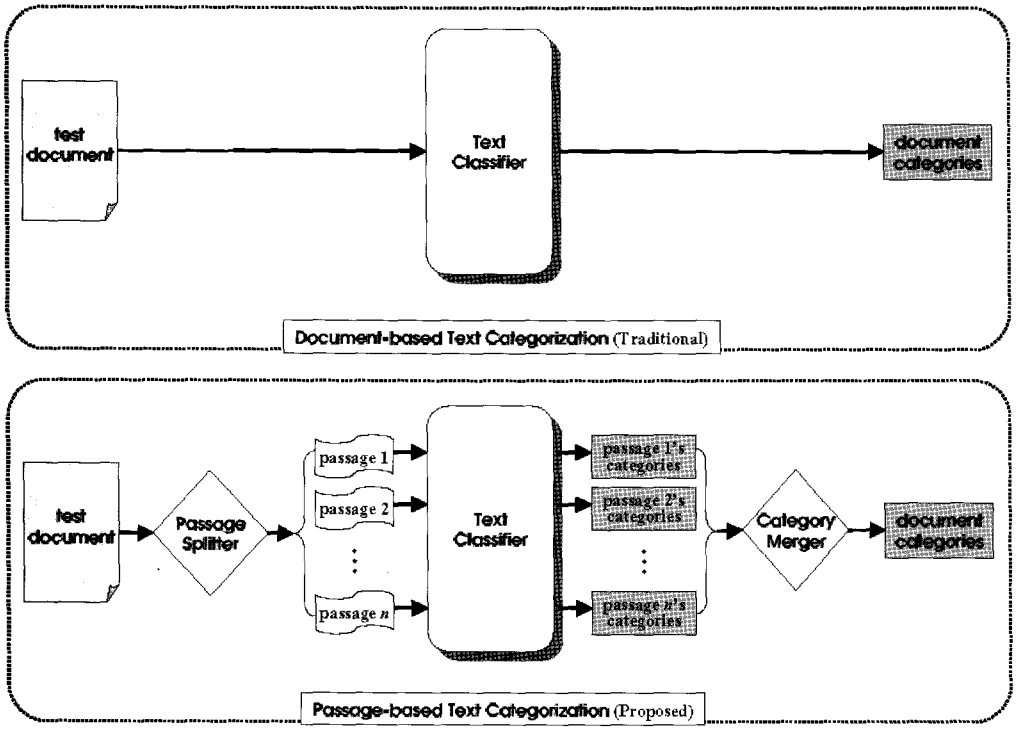
문단의 분리 방법에 따라 여러 가지의 문단타입이 생성되는데, 하위 절에서는 문단에 대한 선행 연구들의 고찰 및 분석을 통해 본 논문에서 비교 대상으로 삼은 여러 가지의 문단 타입들[2]을 도출해 낸다. 문단 타입의 생성 과정에서 사용한 알고리즘은 가장 일반적인 알고리즘으로써 각각에 대해 참조 논문을 언급하였다. 정의한 문단 타입은 모두 문서에 기여하는 바가 다른데, 실험 결과에서 문단 타입에 따른 인자들과 함께 자세히 설명한다.

2.1.1 논리적 문단(Discourse Passage)

논리적 문단은 문장(Sentence)이나 단락(Paragraph)과 같은 문서의 논리적인 요소를 기본으로 하는 직관적인 정의로서[3, 5], 내용에 따라 구성요소들을 조직화 한다. 그러나 논리적 문단에는 다음과 같이 세 가지의 결점이 있다. 1) 문서가 저자들 간의 논리적인 일관성을 보장하지 못한다[3]. 2) 많은 문서들이 문단의 구분 없이 제공되기 때문에, 논리적 문단의 생성이 불가능할 수 있다[7]. 3) 논리적 문단의 길이가 다양하다[7].

1)과 3)에 대한 해결 방안으로서 *경계 단락(Bounded-*

1) 문장(sentence), 단락(paragraph), 절(section), 연속적인 텍스트 조각(segment)
2) 도출 문단 타입들은 이탤릭체로 구분되어 표시하였다.



(그림 1) 문서 단위 범주화와 문단 단위 범주화 모델에 대한 비교

*Paragraph)*이라는 문단 타입을 제안하였다[3]. 경계 단락을 구성하는 동안 짧은 단락들이 순차적인 단락으로 병합되어 단락에 대한 특정 최소 길이를 보장한다. Callan은 경계 단락의 최소 길이로 50 단어를 사용하였다[3].

2.1.2 의미론적 문단(Semantic Passage)

논리적 문단은 원 문서의 구조가 명확하지 못할 때, 일관성이 보장되지 않거나 실용적이지 못하다. 대응 방안으로서 문서를 일정한 주제나 하위 주제에 대응하는 의미론적 문단으로 분리할 수 있다. Kaszkiel은 문서를 이러한 단위들로 분리하는 알고리즘을 제안하고 개발하였다[7]. 이러한 알고리즘 중 *텍스트타일링(TextTiling)*에서는 문서의 하위 주제 구조를 표현하기 위해서 전체 길이 문서를 밀착된 다중 단락 단위로(TextTiles, tiles) 분리하였다[5]. 텍스트타일링에서는 문서를 작은 텍스트 타일(tile)로 분리하고, 단어 빈도(term frequency)를 기반으로 이웃한 타일들의 유사도를 계산한다. 상대적으로 낮은 유사도를 갖는 두 타일은 두 이웃한 타일 간의 경계로서 사용하고, 반면에 높은 유사도를 갖는 두 타일을 하나의 타일로 병합한다.

2.1.3 산술적 문단(Window Passage)

논리적, 의미론적 문단이 문서의 구조적인 특성들에 기초한 반면, 산술적인 문단은 단어들의 순서에 의존한다. 이러한 문단은 단순하며, 구조적인 문서뿐만 아니라 비-구조적인 문서에도 적용될 수 있다.

Hearst와 Plaunt는 문서들을 동일 크기의 블록으로 분리하였는데, 이웃하는 블록들 간에는 공유 부분을 가지지 않도록 하였고, 이러한 문단 타입을 *비중첩윈도우(Non-Overlapping Window)*라 하였다[4]. Callan은 이웃하는 두 세그먼트들이 경계선에서 단어들을 공유하도록 문서를 *중첩 윈도우(Ovelapping Window)*로 분리하였다[3]. 또 다른 윈도우 타입은 *페이지(Page)*로서 경계 단락과 유사한데, 페이지가 물리적인 길이(Bytes)에 따라 구분되는 반면 경계 단락은 단어 수에 의해 구분된다는 차이점이 있다. Moffat은 페이지의 최소 길이로서 1KB를 사용하였다[9].

2.2 문단 범주들로부터 문서 범주를 선택하는 방법

문서안의 문단은 출현 위치에 따라 문서의 주요 주제에 기여하는 바가 다르다. 일반적으로 저자는 자신의 의도를 잘 표현할 수 있도록 조직화된 방법으로 문서를 기술한다. 예를 들면, 신문 기사는 제목에 중요 주제를 두고, 서두 부분에 독자의 주의를 집중시킨다. 반면에 과학적인 기사는 마지막 부분에 결론을 둔다.

제안하는 문서 범주화 모델에서 문서 범주에 대한 문단의 기여 정도는 문단 가중치(Passage Weight)로 정의하는데, 문단 가중치는 문서 내에서의 문단의 출현 위치에 따른 문서 범주화에 대한 기여 정도로 표현 될 수 있다. 3.2절의 <표 1>에 보인 바와 같이, 본 논문에서는 6가지의 문단 가중치 함수(pwf)를 정의하여 사용한다.

문단 범주 병합 과정은 여러 단계를 거치는데, 먼저 분리된 각각의 문단에 대해 가능한 범주들을 할당한다. 문단 가중치 함수에 따라 각각의 문단의 문단 가중치를 계산한다. 한 범주의 가중치는 해당 범주를 부여 받은 모든 문단의 가중치를 합산하여 결정한다. 임계값 이상의 범주 가중치를

가지는 범주들을 해당 문서의 범주로서 결정한다. 문서 범주 결정에 대한 보다 자세한 절차는 3장에서 설명한다.

3. 문서 범주 결정 방법

3.1 kNN 문서 분류기의 사용

문단 기반 범주화 방법을 사용하여 범주를 결정하는 과정에서 문서 기반 범주화에 사용하는 kNN분류기를 사용하였다[14]. kNN은 예제 기반의 분류기[11, 14]로서 전형적인 정보검색 방법과 매우 유사한데, 비교적 느린 분류기로 인식되어 있다. 그러나 기본 정보 검색 시스템과 구현 방법에 따라 분류 속도에 있어 큰 차이가 발생한다. 실험에서 사용한 kNN은 KRISTAL-2002[3]라는 정보 검색 시스템 위에서 효율적으로 개발하였다.

kNN분류기에서는 최상위 k개의 랭킹 된 문서를 얻기 위해서 질의 문서(q)와 대상 문서(d)사이의 유사도($sim(q,d)$)를 측정하는 과정에 벡터 공간 모델을 사용하는데, 수식은 식(1)과 같이 정의한다.

$$Sim(q,d) = \frac{1}{W_t} \sum_{t \in q \wedge d} (w_{q,t} \cdot w_{d,t} + min(f_{d,t}, f_{q,t})) \quad (1)$$

$$W_{d,t} = log(f_{d,t}+1) \cdot log(\frac{N}{f_t}+1)$$

$$W_{q,t} = log(f_{q,t}+1) \cdot log(\frac{N}{f_t}+1)$$

$$W_d = log(\sum_{t \in d} f_{d,t})$$

$f_{x,t}$는 문서 x에서의 단어 t의 빈도이고, N은 전체 문서 수, $min(x,y)$는 x와 y중 작은 값을, f_t는 단어 t가 한번 이상 출현하는 문서 수, $w_{x,t}$는 질의 또는 문서 x에 있는 단어 t의 가중치를, W_d는 문서 d의 길이를 나타낸다.

위의 식(1)은 정보검색에서 자주 사용되는 전형적인 벡터 모델로서, 경험적인 TF · IDF를 기반으로 하여 유도한 것이다[13]. 전형적인 벡터 모델과 비교하여 식(1)의 두드러진 차이점은 질의-문서 유사도 계산 시 $min(f_{d,t},f_{q,t})$ 수식을 사용한다는 점인데, 해당 수식을 사용함으로써 전형적인 벡터 공간 모델에 비해 약간의 성능향상을 볼 수 있었다(실험 데이터 생략). 식(1)의 $min(f_{d,t},f_{q,t})$의 합은 질의와 대상 문서에 동시 출현한 단어의 전체 단어 빈도를 의미하는 것으로써, 유사도 계산 시 상호출현정보(co-occurrence)를 반영한다.

3.2 적합 범주 판정

먼저 문서 단위 범주화에 대해 간략하게 살펴보자. 문서 수준에서 특정 문서 d가 카테고리 c_j ($c_j \in C = \{c_1, c_2, ..., c_{|C|}\}$)에 속하는가를 결정하기 위해서 kNN 분류기를 사용한다. 먼저 문서 d에 가장 유사한 k개의 학습 문서를 획득한 후, 문서 d와 획득된 학습문서 중 c_j에 속하는 문서 사이의 유사도를 합하여 가중치를 계산한다. 가중치가 충분히 크다면 범주에 포함한다. 문서 d에 대한 범주 c_j의 가중치는 **범주 적합도**(category relevance score)로서 $Rel(c_j,d)$로 표현하는데, 다음의 식(2)과 같이 계산한다.

$$Rel(c_j,d) = \sum_{d' \in R_k(d) \cap D_j} Sim(d',d) \quad (2)$$

$R_k(d)$는 문서 d에 가장 유사한 상위 k개의 문서들이고, D_j는 범주 c_j에 할당된 학습 문서들의 집합이고, $Sim(d',d)$는 3.1절의 식(1)에 의해 계산한 문서 간 유사도이다. 각각의 실험 문서는 주어진 임계치 이상의 범주 적합도를 가지는 범주를 할당받는다.

문단 단위 범주화에서는 문서 단위 범주화 작업과 동일한 절차를 사용하여 실험 문서 d의 각각의 문단 p_i에 범주를 할당한다. 문서 d의 범주를 결정하기 위해 식(3)와 같이 문서 d내의 모든 문단의 범주로부터 모든 후보 범주에 대한 적합도를 계산해낸다.

$$Rel(c_j,d) = \sum_{p_i \in P_j} pwf_n(i) \quad (3)$$

p_i는 문서 d의 i번째 문단이고, P_j는 범주 c_j에 할당된 문단 집합이고, $pwf_n()$은 아래의 <표 1>에 설명한 바와 같이 문단 가중치 함수 중의 하나이다. 문서 수준의 범주화와 동일하게 주어진 임계치 이상의 범주 적합도를 가지는 범주들을 문서에 할당한다.

2.2절에서 이미 언급한 바와 같이 6가지의 문단 가중치 함수($pwfs$)를 사용할 수 있는데, 문단 가중치 함수는 문단의 출현 위치를 표현하는 지표로서 0부터 1사이의 실수형 값을 반환한다. 명확한 설명을 위해 <표 1>에서 정규화 인자(Normalization Factor)는 생략한다. 또한 각 문단 가중치 함

<표 1> 문단 가중치 함수

문단 가중치 함수	가중치 성향
$pwf_1(p) = 1$	머리말 = 본문 = 꼬리말
$pwf_2(p) = p^{-1}$	머리말 ≫ 본문 ≫ 꼬리말
$pwf_3(p) = 1p$	머리말 < 본문 < 꼬리말
$pwf_4(p) = \sqrt{(p-\frac{n}{2})^2}$	머리말 = 본문 > 꼬리말
$pwf_5(p) = \sqrt{(\frac{n}{2})^2-(p-\frac{n}{2})^2}$	머리말 = 본문 < 꼬리말
$pwf_6(p) = \frac{1}{log(p+1)}$	머리말 > 본문 > 꼬리말

명확성을 위해 정규화 요소는 생략
p : 문단 위치
n : 전체 문단 수

3) 서지 사항이나, 학위 논문, 저널 기사와 같은 반 정형의 텍스트를 관리하고 검색하기 위해 개발 되었다. 보다 자세한 정보는 http://giis.kisti.re.kr를 참고한다.

수의 성향은 문서를 문단이 아닌 머리말(Head), 본문(Body), 꼬리말(Tail)의 세 부분으로 나누는 경우에 대응하여 해석할 수 있다. 예를 들면, 첫 번째 문단 가중치 함수는 머리말, 본문, 꼬리말이 동일한 비중으로 중요한 것임을 나타낸다. 두 번째 문단 가중치 함수는 머리말이 가장 중요하고 상대적으로 꼬리말보다는 본문이 매우 중요하며 꼬리말에는 큰 의미를 두지 않음을 뜻한다.

4. 데이터 컬렉션(Collection) 및 성능 평가 방법

4.1 데이터 컬렉션

텍스트 범주화에서의 문단의 효과를 입증하기 위해서 로이터(reuter) 버전 3의 세 개의 하위집합 GT800, GT1200, GT1600과 KISTI 학위논문(Theses) 집합을 사용하였다. 본 논문의 실험에 사용한 실험 컬렉션에 대한 전체 현황은 <표 2>에 보인다.

로이터 버전 3 컬렉션은 C.Apte et al.에 의해 구성되었는데[1], 학습 집합과 실험 집합 모두에 라벨이 부여되지 않은 문서들을 제거하였고, 범주가 2이상의 학습 집합 빈도를 갖도록 하였다. 이후부터 구성자의 이름을 따서 데이터 집합의 이름을 '*Apte 컬렉션*'이라 한다.

GTnnnn 테스트 컬렉션은 Apte 데이터 집합에서 문서 길이가 nnnn 바이트 이하인 문서를 제거하였다. 이러한 제약을 이용하여 GT800, GT1200, GT1600 각각에 대해 1109, 652, 410개의 실험 컬렉션을 생성하였다.

마지막으로 KISTI 학위논문 데이터 컬렉션은 KAIST, POSTECH, CNU의 석·박사 학위 논문으로 구성되는데, 20개 학과에 걸쳐 총 1,042개의 문서로 구성되어 있고 대부분의 문서는 한글로 작성되어 있다. 본 논문에서는 20개의 학과를 범주로 선정하고 각 범주 문서의 1/3에 해당하는 347개의 문서를 실험 집합으로 삼아, 2/3에 해당하는 695개의 문서를 학습 집합으로 구성하였다. <표 3>은 실험에 사용한 KISTI 학위논문 데이터 컬렉션 현황을 보이는데, 범주 이름 순으로 정렬되어 있다.

〈표 2〉 실험 컬렉션 현황

컬렉션	Apte	GT800	GT1200	GT1600	KISTI 학위논문
실험 집합	3309	1109	652	410	347
학습 집합	7789	7789	7789	7789	695
범주 수	93	93	93	93	22
최소 텍스트 크기(KB)	0.1	0.8	1.2	1.6	14.8
평균 텍스트 크기(KB)	0.8	1.8	2.2	2.7	92.9

〈표 3〉 KISTI 학위논문 데이터 컬렉션 현황

범주	실험 집합(문서수)	학습 집합(문서수)	실험집합+학습집합
경영학(Management Engineering)	44	90	134
금속 공학(Metal Engineering)	11	17	28
기계공학(Mechanical Engineering)	45	98	143
물리학(Physics)	15	27	42
사회 공학(Civil Engineering)	9	17	26
산업공학(Industrial Engineering)	14	29	43
산업디자인(Industrial Design)	1	4	5
생물학(Biology)	23	39	62
소재 과학(Materials Science)	20	33	53
수학(Mathematics)	7	21	28
응용 소재 공학(Advanced Materials Engineering)	2	5	7
자동화 & 디자인 기술(Automation & Design Technology)	1	4	5
전산학(Computer Science)	24	49	73
전자공학(Electrical Engineering)	48	102	150
정보 통신 공학(Information & Comm. Engineering)	6	15	21
철강 공학(Steel Engineering)	7	12	19
항공우주 공학(Aerospace Engineering)	6	11	17
핵공학(Nuclear Engineering)	7	19	26
화공학(Chemical Engineering)	27	47	74
화학(Chemistry)	25	49	74
환경공학(Environmental Engineering)	5	6	11
기타(Miscellaneous)	0	1	1

〈표 4〉 평균 문단 수와 길이

문단 타입	Apte	GT800	GT1200	GT1600	Theses
비중첩(Non-overlapping) 윈도우	1.8(0.5)	2.2(0.5)	4.2(0.6)	4.7(0.6)	114.0(0.8)
중첩(Overlapping) 윈도우	2.4(0.5)	5.1(0.5)	6.5(0.6)	8.0(0.6)	226.4(0.8)
단락(Paragraph)	7.1(0.1)	10.8(0.2)	12.8(0.2)	15.6(0.2)	90.8(1.0)
경계단락(Bounded-paragraph)	2.3(0.4)	2.8(0.6)	5.4(0.4)	6.6(0.4)	72.3(1.3)
페이지(Page)	N/A	N/A	N/A	N/A	56.9(1.6)
타일(TextTile)	1.9(0.5)	3.1(0.6)	3.5(0.6)	3.8(0.7)	64.4(1.4)

문서 당 평균 문단 수
괄호()안에는 문단의 평균 길이를 KB로 나타냄

4.2 성능 평가 방법

다양한 문단에 대한 범주화 효과를 평가하기 위한 기본 성능 척도로서, 정확도(precision)와 재현율(recall)의 두 개념을 사용하였다. 이 두 개념과 더불어 범주화 성능 측정 시 F_1 방법을 많이 사용하는데, F_1은 정확도와 재현율에 대한 조화 평균의 개념으로 식(4)와 같이 정의한다[12].

$$F_1 = \frac{2pr}{p+r} \tag{4}$$

정확도와 재현율이 동일한 지점의 값을 *break-even point (BeP)*로서 정의하는데, 동일하게 만들 수 없는 경우에는 정확도와 재현율이 가장 가까워질 때의 두 값을 평균한 값을 대신 사용하기도 한다. 이론적으로 BeP는 임의지점의 F_1 수치보다 항상 작거나 같기 때문에 분류기나 범주화 방법들의 성능을 평가하기 위해서 자주 사용한다[11, 16].

본 논문에서 범주화 성능 측정을 위해 정확도, 재현율, BeP의 개념을 우선적으로 사용하는데, BeP 값의 측정이 불가능한 경우에는 F_1 측정 방법을 대신 사용하였다. 범주화에 있어서 범주를 할당받거나 할당받지 못하는 것들 사이의 불균형에 따른 정확성(accuracy)의 오류가 발생한다. 범주들 전반에 걸쳐 정확도와 재현율의 평균값을 사용하기 위해 *micro-averaging* 방법을 사용한다[11].

5. 실험 및 평가 결과

본 논문의 주요 목적은 문단 위치에 따른 문단 가중치 함수와 문단 타입에 따른 문단의 효과의 검증에 있다.

5.1 경험적인 설정

공백 문자로 분리 된 단어들을 문서와 문단의 자질(feature)로서 추출하였다. 수 차례의 휴리스틱한(heuristic) 지식에 의해 범주화 성능의 저하를 막기 위한 스테밍(stemming) 절차는 생략하였는데, Baker와 McCallum의 연구 결과를 참조하기 바란다[2]. 불용어(stopword)는 자질 풀(pool)에서 생략되었다. 또한 숫자 단어를 생략함으로써, Apte와 하위 컬렉션의 경우 약간의 성능 향상을 보였다. KISTI-Theses의 경우 대부분의 문서가 한글로 작성되었기 때문에, 자질 추출 과정에 KISTI가 보유한 한국어 형태소 분석기를 사용하였다.

Yang과 Pedersen(1997)의 실험결과에 의하면, 문서 범주화의 자질 선택 과정에서 문서 빈도를 이용하는 것이 가장 단순하면서 효과적으로 임계치를 설정하는 방법인 것으로 입증되었다[15]. 본 논문에서도 자질 선택 과정에서 문서 빈도(document frequency)를 이용하였다. 문서 수준의 범주화 작업에 대한 자질을 선택하기 위해서 문서 빈도의 범위를 다양하게 설정하여 각각의 테스트 컬렉션에 대해 최상의 결과를 보인 최소 문서빈도와(DF_{min})와 최대 문서 빈도(DF_{max})를 구하였고, 동일한 값을 문단 수준의 범주화에 사용하였다.

kNN 분류기의 경우, 문서 수준 범주화에서 가장 적합한 k값을 문단 수준 범주화에 동일하게 적용하였다. Apte, GT800, GT1200, 1600 데이터 집합의 경우, k값으로 10을 사용하였고, KISTI-Theses 컬렉션의 경우 k값으로 1을 사용하였다. KISTI-Theses의 경우, 각 문서의 길이가 선택된 범주를 설명하기에 충분한 길이를 갖기 때문에 각각의 문서에 단 하나의 범주만을 할당하였다.

아래의 <표 4>는 데이터 집합 안의 실험 문서의 평균 문단 수와 평균 문서 길이를 보여준다. 비중첩 윈도우(Non-overlapping Window)와 중첩 윈도우(Overlapping Window)의 경우 문단 길이는 100 단어로 제한하였다. 페이지 길이는 최소 1KB로 정하였지만, 1KB 또는 2KB처럼 짧은 문서에 페이지(Page) 타입을 적용시키는 것은 무의미하다고 판단하였기 때문에, Apte와 하위 3가지 컬렉션의 경우 페이지 타입의 효과를 실험하지 않았다.

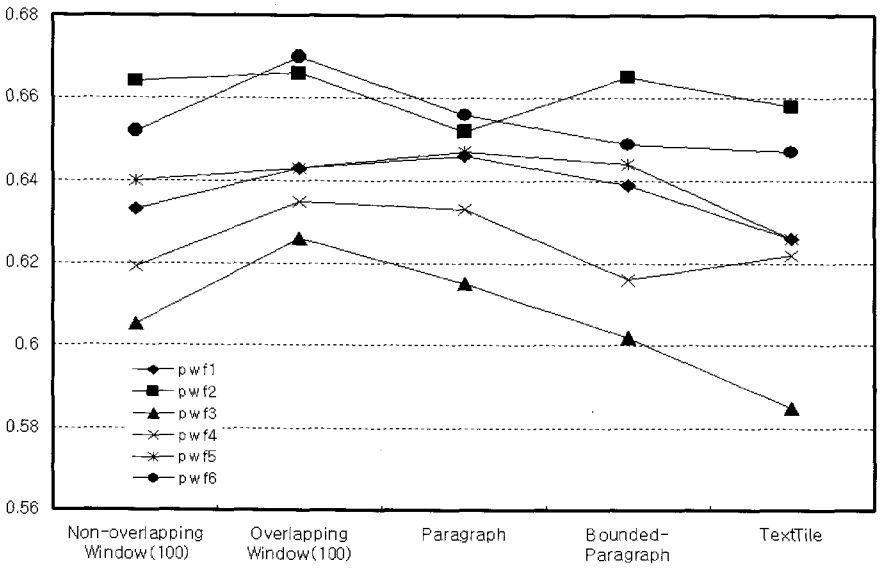
5.2 문단 가중치 함수의 효과

2.2절에서 언급한 바와 같이 문단의 위치는 문서의 주요 주제를 결정하는데 있어서 중요한 역할을 한다. 이러한 점을 가정으로 하여 6개의 다른 종류의 문단 가중치 함수를 소개하는데, 함수는 문단 위치를 표시하는 지표로서 0부터 1 사이의 값을 반환한다(<표 1> 참조). 이러한 문단 가중치를 합산함으로서 문서의 범주를 결정한다(그림 1 참조).

GT1600 컬렉션에 대한 BeP값은 6개의 문단 가중치 함수를 모두 고려하여 (그림 2)에 보인다.

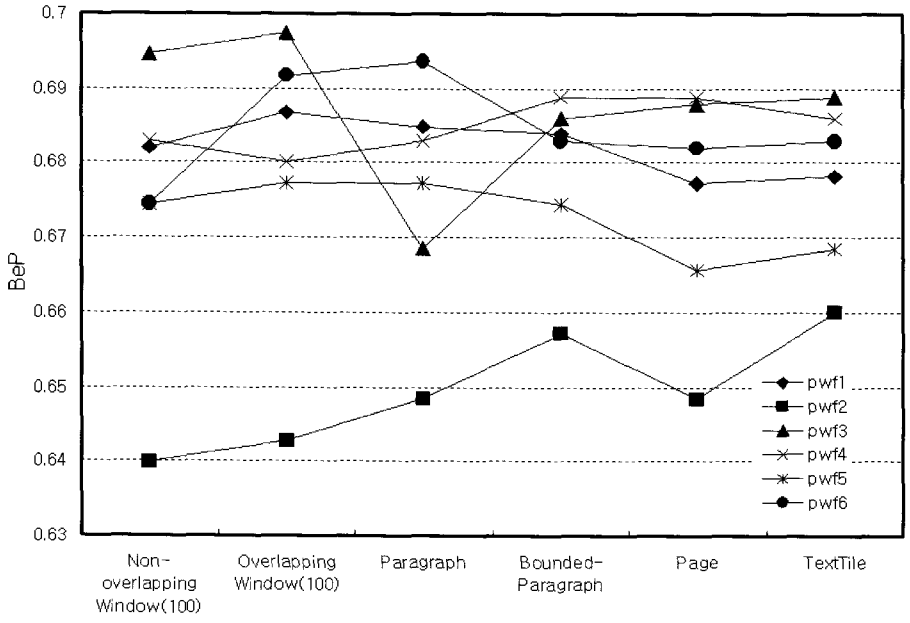
GT1600 컬렉션에서 문단 가중치 함수 pwf2와 pwf6이 가장 좋은 성능을 보이는데, 이 함수들의 경우 머리말(Head로)부터 높은 문단 가중치를 받고, 본문(Body)에서는 중간

(데이터 집합: GT1600)



(그림 2) 6가지의 문단 가중치 함수의 효과

(데이터 집합 : KISTI-Theses)



(그림 3) 6가지 문단 가중치 함수의 효과

정도, 꼬리말(Tail)에서는 낮은 가중치를 받는 것에 기인하는 것으로 분석됐다. pwf3은 정 반대의 경우로서 가장 좋지 않은 성능을 보인다. Apte, GT800, GT1200 컬렉션의 경우도 동일한 성능을 보이고 있다. 이러한 결과는 로이터 뉴스 기사들의 주요 주제가 문서의 처음 부분에 의해 결정됨을 단적으로 보여준다.

(그림 3)을 참조하면 KISTI-Theses 데이터 집합의 경우에는 이전 결과와 매우 다른 성향을 보이는데, pwf2가 모든 문단 타입에 대해 최하 성능을 보인다. 반면에 pwf1, pwf3, pwf6은 비슷한 성능을 보이는데, pwf1 보다는 pwf3이나 pwf6의 성능이 높다. 이러한 현상은 KISTI-Theses 데이터 집합이 논문으로 구성되어 있다는 특성인 기인한다. 주요 주제는 특정한 부분보다 머리말, 본문, 꼬리말에 고루 분포하고 있고, 본문 보다는 머리말이나 꼬리말에 더 많은 주요 주제가 위치함을 단적으로 보여준다.

pwf1은 문단의 위치와 무관하게 1값을 반환하지만, 다른 문단 가중치 함수들은 문서 내의 문단의 위치에 따라 다른 값을 반환한다. (그림 2)에서 pwf2와 pwf6이 pwf1에 비해 높은 성능 향상을 보인다는 점과 (그림 3)에서 pwf3과 pwf6이 pwf1에 비해 높은 성능 향상을 보인다는 점은 각각의 문단이 문서의 주요 주제에 기여하는 바가 다르다는 가정을 뒷받침해 준다. 이것은 문서 도메인에 따라 다른 문단 가중

치 함수를 적용해야 한다는 사실을 분명히 하고 있다.

문단 가중치 함수의 효과에 대한 실험을 통하여 Apte 계열과 KISTI-Theses 컬렉션 각각의 경우, 효율적인 문단 가중치 함수가 pwf2와 pwf3임을 알 수 있었다. 다음 절의 문단의 효과에 대한 실험에서는 이미 발견한 가중치 함수를 사용하여 실험을 수행하였다.

5.3 문단의 효과(문단 타입의 효과)

5.3.1 Apte 컬렉션

5.2절에서 설명한 바와 같이 가장 우수한 성능을 보여주는 문단 가중치 함수 pwf2를 사용하였다. Apte 컬렉션에 대한 실험 결과는 다음의 <표 5>에서 볼 수 있는데, 가장 마지막 열은 문서 기반의 범주화 모델과 비교한 성능 향상 정도를 %로 표현한 것이고, 밑줄로 표현된 부분은 가장 좋은 성능을 보일 때의 BeP값을 나타낸다.

<표 2>에서 볼 수 있듯이, Apte, GT800, GT1200, GT1600 테스트 컬렉션은 비교적 짧은 문서로 구성되어 있기 때문에 문단 타입 중 페이지 타입을 적용하지 않았다. 또한 표준 텍스트타일링 알고리즘은 매우 큰 형태의 타일을 생성하는 알고리즘이기 때문에, 본 논문에서 사용하는 데이터 집합에 적용시키기에는 무리가 있었다. 따라서 알맞은 덩어리의 타일 크기를 보장하도록 텍스트타일링 알고리즘을 수정하여 사용하였는데, 표준 텍스트타일링 알고리즘을 사용할 때 타일의 평균 크기는 1내지 2KB인데 비해 수정된 알고리즘에서 타일의 평균 크기는 98단어에 454 Bytes정도로 매우 작았다(<표 4> 참조).

BeP상으로 크게 성능 향상을 가져오지는 않았지만, 중첩 윈도우를 사용할 경우, 문단 길이 100과 150 단어에서 가장 좋은 성능을 보였다. 반면 경계 단락(Bounded-Paragraphs)의 경우에는 최소 50에서 최대 200단어를 포함하는 단락으로서, 최악의 성능을 보였다[3]. Apte 컬렉션은 비교적 문서 길이가 짧기 때문에 경계단락의 형성 과정에서 오류가 발생하여 예기치 않은 성능 저하를 보이는 것으로 추정된다.

〈표 5〉 Apte 데이터 컬렉션에 대한 문단 효과(pwf2 적용)

문단 타입	Precision	Recall	BeP+	△%
문서 단위	0.818	0.818	0.818	0.0
비중첩 윈도우				
window size = 50	0.817	0.817	0.817	-0.1
window size = 100	0.820	0.821	0.820	0.3
window size = 150	0.822	0.822	0.822	0.5
window size = 200	0.819	0.824	0.822∓	0.4
중첩 윈도우				
window/overlaps = 50/25	0.820	0.820	0.820	0.3
window/overlaps = 100/50	0.823	0.823	0.823	0.6
window/overlaps = 150/75	0.823	0.823	0.823	0.6
window/overlaps = 200/100	0.822	0.822	0.822	0.4
단락	0.812	0.812	0.812	-0.8
경계단락	0.761	0.766	0.763∓	-6.7
타일	0.831	0.812	0.821∓	0.4

+ Micro-averaged precision recall 과 break-even points

∓ Micro-averaged F_1 방법

실험결과 데이터 집합에서 모든 문단 타입에 대해서 성능 향상이 미미한 이유는 짧은 문서의 비율이 매우 높기 때문이었다. 예를 들면, 실험 집합에서 문서의 약 60%가량은 100 단어 이하로 구성되고 윈도우 크기가 100인 한 개의 비중첩 윈도우를 생성한다. 이러한 특성으로 인해 문단 수준의 범주화에 의한 성능 향상이 있더라도, (그림 1)에서 보인 문단 범주의 병합 과정을 제대로 수행할 수 없어서 성능 향상 정도가 미미한 것으로 확인됐다.

실험 문서의 길이가 작음으로 인한 부정적인 효과를 제거하고 문단 수준 범주화의 효과를 검증하기 위해서, Apte의 하위 집합 GT800, GT1200, GT1600을 사용한 결과를 다음 절들에서 설명한다.

5.3.2 GT800 컬렉션

GT800은 Apte 실험 컬렉션의 하위 집합으로서 800바이트 보다 작은 문서들을 제거한 실험 집합이다. GT800에 대한 실험 결과는 다음의 <표 6>에 보인다. 대부분의 문단 타입에 대해서 성능 향상을 보였는데, 그 중 중첩 윈도우의 크기가 100일 경우 가장 좋은 성능을 보였다.

〈표 6〉 GT800 데이터 컬렉션에 대한 문단의 영향(pwf2 적용)

문단 타입	Precision	Recall	BeP+	△%
문서 단위	0.690	0.690	0.690	0.0
비중첩 윈도우				
window size = 50	0.686	0.688	0.688	-0.3
window size = 100	0.695	0.690	0.697	1.0
window size = 150	0.706	0.701	0.703	1.9
window size = 200	0.706	0.706	0.706	2.3
중첩 윈도우				
window/overlaps = 50/25	0.690	0.690	0.690	0.0
window/overlaps = 100/50	0.715	0.707	0.711∓	3.0
window/overlaps = 150/75	0.704	0.704	0.704	2.0
window/overlaps = 200/100	0.706	0.710	0.708	2.6
단락	0.696	0.697	0.697	0.9
경계 단락	0.688	0.707	0.697∓	1.0
타일	0.704	0.702	0.703	1.9

+ Micro-averaged precision recall 과 break-even points

∓ Micro-averaged F_1 방법

5.3.3 GT1200 컬렉션

GT1200 컬렉션은 apte컬렉션 중에서 문서 길이가 1200 이하인 것을 제거하여 재구성한 실험 집합이다. Apte와 동일한 환경에서 실험하였고, GT1200 실험 컬렉션에 대한 실험 결과는 <표 7>에 보인다. 대부분의 문단 타입에 대해서 성능 향상을 보였는데, 그 중 중첩 윈도우의 크기가 100일 경우 가장 좋은 성능을 보였다. Apte와 GT800을 비교해보면 실험 집합의 크기가 증가함에 따라 문단 수준 범주화와 문서 수준 범주화 사이의 BeP 차별이 명확해 졌다.

〈표 7〉 GT1200 데이터 컬렉션에 대한 문단의 영향(pwf2 적용)

문단 타입	Precision	Recall	BeP+	△%
문단 단위	0.660	0.660	0.660	0.0
비중첩 윈도우				
window size = 50	0.674	0.673	0.673	2.1
window size = 100	0.689	0.683	0.686∓	4.0
window size = 150	0.665	0.664	0.665	0.8
window size = 200	0.642	0.637	0.640∓	-3.0
중첩 윈도우				
window/overlaps = 50/25	0.677	0.678	0.678	2.7
window/overlaps = 100/50	0.690	0.689	0.689	4.5
window/overlaps = 150/75	0.665	0.664	0.665	0.8
window/overlaps = 200/100	0.643	0.645	0.644	-2.3
단락	0.675	0.675	0.675	2.3
경계 단락	0.683	0.683	0.683	3.5
타일	0.671	0.670	0.671	1.7

+ Micro-averaged precision recall과 break-even points
∓ Micro-averaged F_1 방법

〈표 8〉 GT1600 데이터 컬렉션에 대한 문단의 영향(pwf2 적용)

문단 타입	Precision	Recall	BeP+	△%
문서 단위	0.636	0.636	0.636	0.0
비중첩 윈도우				
window size = 50	0.649	0.648	0.648	1.9
window size = 100	0.665	0.663	0.664	4.4
window size = 150	0.653	0.642	0.647∓	1.8
window size = 200	0.626	0.621	0.623	-2.0
중첩 윈도우				
window/overlaps = 50/25	0.658	0.658	0.658	3.4
window/overlaps = 100/50	0.670	0.670	0.670	5.4
window/overlaps = 150/75	0.669	0.663	0.666	4.6
window/overlaps = 200/100	0.649	0.649	0.649	2.0
단락	0.652	0.652	0.652	.2.5
경계 단락	0.665	0.665	0.665	4.5
타일	0.659	0.658	0.658	3.4

+ Micro-averaged precision recall과 break-even points
∓ Micro-averaged F_1 방법

5.3.4 GT1600 컬렉션

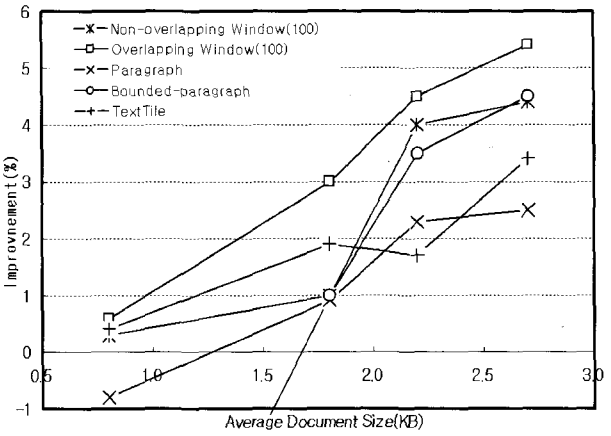
GT1600 컬렉션은 apte 컬렉션 중에서 문서 길이가 1600 바이트 이하인 것을 제거하여 재구성한 실험 집합이다. Apte와 동일한 환경에서 실험하였고, 그에 대한 결과는 다음의 <표 8>에 보인다.

Apte와 모든 GTnnnn에 대하여 중첩 윈도우 방식이 가장 좋은 결과를 보였다. 중첩 윈도우 방식은 균일한 문단 길이를 보이는 반면 다른 방식들에서는 문단 길이가 가변적이기 때문에 올바른 문단 범주를 찾지 못하여, 타 방식에 비해 성능 저하를 보인 것으로 추정된다. 더욱이 중첩 윈도우는 모든 데이터 집합에 대한 실험 결과에서 비중첩 윈도우 방식보다 높은 성능 향상을 보였다. 비중첩 윈도우를 포함하는 다른 문단 타입들은 경계선 상에서의 단어 지역 정보(term locality information)를 잃는 경향을 보이는 반면, 중첩 윈도우 방식에서의 문단들은 이웃한 문단과 중첩되기 때문에 지역 정보를 보장하고 있음을 알 수 있다.

가장 좋은 성능을 보였을 때의 문단 길이는 중첩과 비중첩 윈도우 방식의 경우 모두 100 단어인 것으로 나타났고, 이때 단어 수에 수치 값을 가지는 단어는 포함하지 않았다. 100단어는 Apte와 GTnnnn 실험 집합에서 평균적으로 3개의 단락에 해당한다.

<표 6, 7, 8>의 실험 결과에서 문서 대비 문단 범주화의 성능 향상 정도는 문서 길이에 비례함을 알 수 있다. 자세한 설명은 (그림 4)에 설명되어 있다.

(그림 4)에서 성능 향상 정도는 %로 표시하였고, 대상 평균 문서 길이(average document size)는 Apte, GT800, GT1200, GT1600 데이터 컬렉션의 실험 문서들의 평균 길이로 표시하였다. 중첩 문단 타입의 경우 평균 실험 문서의 길이와 성능 향상 사이에는 보다 강한 선형적인 상관관계가 있음을 알 수 있다.



(그림 4) 문서 길이와 성능 향상 사이의 상관관계[4]

지금까지의 실험은 비교적 작은 데이터 컬렉션 내에서의 큰 문서들을 대상으로 실험을 하였다. 다음절에서는 석·박사 학위 논문 전문 데이터 컬렉션에 대한 문단 수준 분류 결과에 대해 설명한다.

5.3.5 KISTI 학위 논문 컬렉션

지금까지 비교적 짧은 문서를 가진 실험 집합을 대상으로 문단 단위 문서 범주화를 실험하였다(<표 2> 참조). 이 절에서는 평균 문서 길이 92.9KB(최소 : 14.8KB, 최대 : 535.5KB)를 보이는 장문 문서로 구성된 KISTI 학위 논문 실험 집합을 대상으로 한 결과를 제시한다. 문단 가중치 함수 3(pwf3)을 적용시킨 결과를 <표 9>에 보인다.

실험에서 kNN 분류기에 대한 k 값으로 1을 선택하였다. 문서 빈도가 2에서 69사이의 것을 자질 선택 기준으로 삼았는데, 이것은 학습 문서들의 10%에 해당된다. 컬렉션의 크

4) 경계 단락의 경우는 매우 큰 바이어스로 인해 첫 꼭지 값을 생략하였다.

〈표 9〉 KISTI 학위 논문 데이터 컬렉션에 대한 문단의 영향 (pwf3 적용)

문단 타입	Precision	Recall	BeP+	△%
문서 단위	0.631	0.631	0.631	0.0
비중첩 윈도우				
window size = 50	0.677	0.677	0.677	7.3
window size = 100	0.695	0.695	0.695	10.0
window size = 200	0.683	0.683	0.683	8.2
window size = 400	0.687	0.689	0.688∓	9.0
중첩 윈도우				
window/overlaps = 50/25	0.669	0.669	0.669	5.9
window/overlaps = 100/50	0.697	0.697	0.697	10.5
window/overlaps = 200/100	0.692	0.692	0.692	9.6
window/overlaps = 400/200	0.686	0.686	0.686	8.7
단락	0.669	0.6669	0.669	5.9
경계 단락	0.686	0.686	0.686	8.7
페이지	0.689	0.689	0.689	9.1
타일	0.689	0.689	0.689	9.1

+ Micro-averaged precision recall과 break-even points

∓ Micro-averaged F_1 방법

기가 비교적 작고 각각의 문서에는 1개의 범주만이 할당되어 있기 때문에 k=1을 사용하였다. 또한 k값이 커짐에 따라 적합하지 않은 문서가 상위 순위에 위치함으로써 성능 저하를 야기한다는 점과 컬렉션 내의 모든 문서의 길이가 충분히 커서(평균 92,900 글자) 한 범주의 자질로서 충분한 단어를 포함하고 있다는 것을 의미한다.

문서 전체를 하나의 단위로 사용할 때보다 임의의 문단을 사용할 때, 범주화 성능이 좋은 것으로 나타났다. (그림 4)와 <표 9>를 참조하면, 모든 실험 집합에 대해서 중첩 윈도우의 문단 길이가 100일 때 가장 좋은 성능을 보였다. 또한 Apte를 제외한 다른 실험 집합의 경우, 경계 단락이 단락 타입에 비해 보다 좋은 결과를 보였다. 2.1절에서 명시한 바와 같이 경계 단락은 일정 최소 길이를 보장하는 반면 일반 단락은 한 문장에서 수 십 문장까지 그 길이가 다양하기 때문에, 경계 단락은 일반 단락에 비해 보다 적게 왜곡되었다. 문단 길이에 있어서의 이러한 왜곡은 문단 단위 문서 범주화에 좋지 않은 영향을 미치는 것으로 나타났다.

마지막으로 KISTI 데이터 컬렉션의 범주화에 있어서 속도(speed)-성능(effectiveness)은 상호보완(trade-off) 관계에 있다. 실험에서 문단 단위 범주화는 문서 단위 범주화에 비해 전체 문서 분류 시간에서 약 2-5배 정도 느린 것으로 나타났다. 반면에 메모리 사용량은 급격히 감소되었는데, 이러한 현상은 문서가 문단 단위로 분리됨으로써, 문단 수만큼 범주화 작업수가 증가되었지만 복잡도가 문서에서 문단으로 줄어든 것에 기인한다. <표 4>에 따르면, KISTI 논문 컬렉션의 경우 하나의 문서는 수백 개의 문단으로 분리되는 경향을 보인다. 더욱이 KISTI-Theses나, 전자책, 전체 웹 사이들과 같이 덩치가 큰 것들에 전체 문서 단위 범주화를 적용하는 것은 비실용적이기 속도-메모리 상호보완 관계를 살펴보아야 한다.

5. 결론 및 향후 연구과제

전문 데이터베이스의 등장과 웹 사이트가 확장됨에 따라, 종전의 전문 문서 분류는 새로운 문서 타입에 부적합하기 때문에 새로운 문서 분류 방법이 필요하게 되었다. 본 논문에서는 새로운 문서 범주화 모델로서 *"문단 단위 문서 범주화 모델"*을 소개하였는데, 가장 큰 차이점은 범주화 단위가 전체 문서에서 문서 내의 문단 단위로 축소되었다는 점이다.

새로운 문단 타입(비-중첩윈도우, 중첩윈도우, 단락, 경계 단락, 페이지, 타일)을 사용하는 문서 범주화를 시도하였다. 문단 수준의 범주화를 이용한 성능 향상 정도를 전체 문서 범주화와 비교해 볼 때, 짧은 문서 컬렉션에 대해서는 5% 이상, 긴 문서 컬렉션에 대해서는 10% 이상의 성능향상을 보였다. 모든 문단 타입에 대해서 범주화 성능향상이 있었다. 그러나 다섯 개의 테스트 컬렉션을 통틀어 다른 문단 타입에 비해 *중첩 윈도우*를 사용하여 월등한 성능 향상을 볼 수 있었다.

문단 범주를 문서 범주로 병합하는 과정에서 *문단 가중치 (passage weight)*를 소개하였다. 본 논문에서 실험된 문단 가중치 방법은 모두 성능 향상을 보였는데, 그 중 단순 합산 가중치 방법(pwf1)에 비해 다른 것들을 이용할 때 월등한 성능 향상을 보았다. 데이터 집합에 따라 최적의 가중치 방법이 다른 것으로 나타났으며, 가중치 방법에 대한 세심한 고려로 보다 나은 성능 향상을 꾀할 수 있을 것으로 보인다.

전체 문서와 달리 문단 단위의 효과를 다음과 같이 정리할 수 있다. 첫째, 문단은 한 문서의 부주제 구조를 대변하는 반면, 전체 문서는 이러한 문서의 구조를 무시하는 경향이 있다. 문단 수준의 문서 범주화에서서 이러한 부주제 구조는 분류기 풀을 사용함으로써 가능한데 풀에는 지역적인 문단 수준의 뷰를 포함한다. 문서 수준의 분류에서는 단어의 지역적인 정보가 사라지지만, 문서를 문단으로 분리함으로써 부분 지역성을 보장한다. 이런 측면에서, 경계 지역에서라도 지역성을 보장하는 중첩 윈도우가 다른 문단 타입에 비해 성능이 우수한 것은 당연하다.

문서를 작은 문단 단위로 분리함으로써 긴 문서를 대상으로 하는 텍스트 범주화에 효과적으로 사용될 수 있다. 짧은 문서 컬렉션의 경우에도 문단을 사용함으로서 어느 정도의 성능 향상을 볼 수 있었다. 문단 단위의 텍스트 범주화가 속도-효과 측면에서 상호보완 관계에 있지만, 긴 문서에 대해서는 이러한 방식을 사용하는 것이 바람직할 것으로 생각된다.

문단 단위 문서 범주화 모델은 비구조적인 전문, XML 문서, 전체 웹 사이트 모든 경우에 효과적인 방법 중의 하나가 될 수 있다. 비구조적인 전문의 경우 논문에서 제시한 문단 타입을 범주화 과정에 사용하여 그 효과를 입증하였다. XML 문서는 자연적으로 문단 단위로 구성되어 있는데, 즉 적절한 엘리먼트(element) 내용을 문단으로 간주한다면 XML 문서에 대해서도 문단 단위 범주화 방식이 적합할 것으로

생각한다. 또한 웹 사이트 분류의 경우에도 하나의 웹 사이트는 다수의 웹 페이지들로 구성되기 때문에, 각각의 웹 페이지를 문단으로 간주할 수 있을 것이다.

문단 단위 문서 범주화 모델은 두 가지 측면에서 다중분류기(classifier committees)와 유사하다[8, 11]. 첫째, 분류의 결과는 분류기 풀에 의해 결정된다. 둘째, 문단 단위 문서 범주화 모델은 문서의 범주를 결정하기 위해서 문단 가중치 함수를 사용한다. 반면에 다중 분류기는 범주를 선택하기 위해 조합 함수를 사용한다[8, 11]. 따라서 다중 분류기의 많은 연구 결과들이 문단 기반의 범주화에 적용될 수 있을 것으로 생각한다.

참 고 문 헌

[1] Apte, C., Damerau, F., and Weiss, F. "Towards Language Independent Automated Learning of Text Categorization Models," Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp.23-30, 1994.

[2] Baker, L. D. and McCallum, A. K. "Distributional Clustering of Words for Text Classification," Proceedings of the 21th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp.96-103, 1998.

[3] Callan, J. P. "Passage Retrieval Evidence in Document Retrieval," Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp.302-310, 1994.

[4] Hearst, M. A., and Plaunt, C. "Subtopic Structuring for Full-length Document Access," Proceedings of the 16th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp.59-68, 1993.

[5] Hearst, M. A. "Multi-paragraph Segmentation of Expository Texts," Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp.9-16, 1994.

[6] Kaszkiel, M., Zobel, J. and Sacks-Davis, R. "Efficient Passage Ranking for Document Databases," ACM Transactions on Information Systems, Vol.17, No.4, pp.406-439, 1999.

[7] Kaszkiel, M., and Zobel, J. "Effective Ranking with Arbitary Passages," The Journal of American Society for Information Science and Technology, Vol.52, No.4, pp.344-364, 2001.

[8] Larkey, L. S., and Croft, W. B. "Combining Classifiers in Text Categorization," Proceedings of SIGIR-96, 19th ACM International Conference on research and Development in Information Retrieval, pp.289-297, 1996.

[9] Moffat, A., Sacks-Davis, R., Wilkinson, R. and Zobel, J. "Retrieval of Partial Documents," NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC 2), pp.181-190, 1994.

[10] Salton, G., Allan, J., and Buckley, C. "Approaches to Passage Retrieval in Full Text Information Systems," Proceedings of the 16th Annual International Conference on Research and Development in Information Retrieval, pp.49-58, 1993.

[11] Sebastiani, F. "Machine Learning in Automated Text Categorization," ACM Computing Surveys, Vol.34, No.1, pp.1-47, 2002.

[12] van Rijsbergen, C. "Information Retrieval," Butter- worths, London, 1979.

[13] Witten, I. H., Moffat, A., and Bell, T. C. "Managing Gigabytes: Compressing and Indexing Documents and Images," Morgan Kaufmann Publishing, San Francisco, 1999.

[14] Yang, Y. "Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval," Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp.13-22, 1994.

[15] Yang, Y. and Pedersen, J.~O. "A Comparative Study on Feature Selection in Text Categorization," Proceedings of the 14th International Conference on Machine Learning (ICML'97), pp.412-420, 1997.

[16] Yang, Y. "An Evaluation of Statistical Approaches to Text Categorization," Journal of Information Retrieval, Vol.1, No.1, pp.67-88, 1999.

[17] Yang, Y., Slattery, S., and Ghani, R. "A Study of Approaches to Hypertext Categorization," Journal of Intelligent Information Systems, Vol.17, No.2, pp.219-241, 2002.

[18] Zobel. J., Moffat, A. Wilkinson, R., and Sacks-Davis, R. "Efficient Retrieval of Partial Documents," Information Processing and Management, Vol.31, No.3, pp.361-377, 1995.

[19] Callan J. Characteristics of text, 1997.

[20] Harman D. "The DARPA Tipster Project," SIGIR Forum, Vol.26, No.2, pp.26-28, 1992.

[21] Kaszkiel M., Zobel J., Davis Sacks-R. "Efficient passage ranking for document databases," ACM Transactions on information systems, Vol.17, No.4, pp.406-439, 1999.

[22] Wilkinson R. "Effective Retrieval of structured documents," Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994.

[23] Ludovic D., Hugo Z. "HMM-based Passage Models for Document Classification and Ranking," Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research, 2001.

[24] Cehn W., Chang X., Wang H., Zhu J., and Yao T. "Automatic Word Clustering for Text Categorization Using Global Information," Proceedings of AIRS 2004.

[25] Thanaruk T. "Applying passage in Web text mining," International Journal of Intelligent Systems, Vol.19, Issue 1-2, pp.149-158, 2004.

주 원 균

e-mail : joo@kisti.re.kr
1997년 충남대학교 전산학과(학사)
1999년 충남대학교 컴퓨터과학과(석사)
1999년~현재 한국과학기술정보연구원 연구원
관심분야 : 정보검색, XML 저장 및 관리

김 진 숙

e-mail : jinsuk@kisti.re.kr
1993년 KAIST 생물과학과(학사)
1995년 KAIST 생명과학과(석사)
2002년 KAIST 전산학과(석사)
1995년~2001년 한국과학기술정보연구원 연구원
2001년~2002년 (주)서치솔루션 연구원
2002년~현재 한국과학기술정보연구원 선임연구원
관심분야 : 정보검색 및 관리, 자동문서분류, 생물서열처리

최 기 석

e-mail : choi@kisti.re.kr
1988년 서울대학교 전산학과(학사)
1997년 KAIST 전산학과(석사)
1988년~2004년 한국과학기술정보연구원 선임연구원
2005년~현재 한국과학기술정보연구원 국가RnD시스템개발실 실장
관심분야 : 데이터베이스, 지식 및 데이터 공학