코호넨 맵에서 태그 정보를 이용한 XML 문서 클러스터링

박사준*, 박현근**

요약

본 논문에서는 XML 문서에 대해서 중요한 특징 중 하나인 임의의 태그 생성 기법을 이용하여 클러스터링한다. 태그 특징 벡터와 문단의 단어를 분리하여 특징 벡터를 생성하여 코호넨 맵에 적용하여 클러스터링을 수행하였다. 태그는 꼭 필요한 키워드이므로 이진법을 사용하고 단어는 TF/IDF 기법을 사용하였다. Reuter-21578 문헌 집합을 이용하여 실험한 결과 50% 전후의 재현률과 정확률을 산출하였다. 또한, 전통적인 자동 문서 분류 알고리즘인 SVM과 K-NN과 비교 실험도 수행하였다. 카테고리가 정해진 분류 시스템인 SVM과 K_NN 시스템과의 비교에서 전체적으로 10%정도 성능이 좋게 나왔다.

Clustering of XML Documents using Tag Information with Kohonen Map

Sa-Joon Park*, Hyun-Gun Park**

ABSTRACT

One of the important features for the XML document is the creation of arbitrary tags. In this paper, we make use of it for clustering XML documents. Tag feature vector and word feature vector are separately created . Clustering was performed by applying a Kohonen map. Because tags are necessary keywords, we utilized binary method for them. TF / IDF technique was used for word feature vector. Reuter-21578 collections are experimented. The results of experimentation is almost rate of 50% in recall and precision rate. In addition, the traditional classification algorithm, SVM and K-NN was also compared with our system. Performance of our results were 10% more than SVM, K_NN system.

Key Words : Clustering, Kohonen Network, XML, tag, Feature Vector

* 대구한의대학교 모바일콘텐츠학부
** 숭실대학교 전산원 e비즈니스경영학과
· 제1저자(First Author) : 박사준 · 교신저자(Correspondent Author) : 박현근
· 접수일(2010년 2월 8일), 수정일(1차 : 2010년 3월 9일), 게재확정일(2010년 3월 12일)

Ⅰ. 서 론

인터넷의 발달로 웹에는 수많은 종류의 문서가 매일 생산되고 있다. 월드 와이드 웹(WWW)에서 XML 문서의 비중은 날로 높아가고 있다. 날로 늘어가는 XML 문서를 같은 종류의 문서끼리 클러스터링하는 방법은 웹 카테고리화, 웹 검색을 위해서 필요한 기술이다. XML는 HTML과 다르게 정해진 태그가 아닌 사용자가 임의로 정의 가능한 태그를 사용하고 있다. 그러므로, 태그는 문서내의 어떤 단어보다도 아주 중요한 정보를 가진 키워드이다.

문서내에서 생성되는 단어보다도 중요도와 정보의 함축성이 내포된 문서의 의미를 결정할 수 있는 정보이다. 본 논문에서는 XML이 가지는 자유로운 태그 생성과 태그의 중요성에 관심을 갖고 있다. XML의 태그 정보를 이용하여 XML 문서를 클러스터링하고, 이를 위해서 코호넨 맵을 이용하여 실험한다.

본 논문에서는 문서에 대한 클러스터링을 XML 문서의 유용한 태그 정보를 이용하여 특징 벡터를 만들고, 특징 벡터를 코호넨 맵에 적용하여 고차원의 문서 데이터를 2차원의 특징 맵에 매핑하고, 타시스템과 비교 실험한다.

문서 클러스터링을 위해서 문헌 집합으로는 1987년의 로이터 뉴스 기사 모음인 Reuters-21578 문헌 집합을 이용하고, 포터(Porter)의 스테밍(stemming) 알고리즘을 이용하여 문서 벡터를 구성하여 실험하였다.

Ⅱ. 기반 연구

문서 분류 방법으로는 SVM(Support Vector Machine)[1][2], K-NN(K-Nearest Neighbor)[3], 베이의 분류자(Naive Bayes Classifier)[4] 등의 방법들이 존재하고 있다.

2.1 SVM(Support Vector Machine)

SVM은 1995년 Vapnik에 제안한 방법[1][2]으로 고차원 특징 공간에서 주어진 양과 음의 학습 데이터를 분류하는 최적의 분류 경계면을 찾는 알고리즘이다.

SVM의 기본 원리는 선형 분리 가능한 문제에서 출발한다. 학습 데이터의 출력으로 {+1, -1}처럼 클래스 구분이 가능하다면 학습 데이터를 두 집합으로 구분할 수 있다.

이때 두 집합으로 구분 짓는 경계면인 초평면(hyperplane)을 찾으면 된다. 초평면은 다음과 같이 표현된다.

$$w^T x + b = 0 \quad (1)$$

여기에서, x는 입력 벡터이고, w는 가중치 벡터, b는 바이어스(bias)이다. 초평면이 학습 데이터의 분류 함수로 주어지면, 가장 완전한 초평면을 구하기 위해서 학습 데이터와의 거리가 가장 떨어진 분류 여백(seperation margin)이 가장 큰 초평면을 구하면 된다.

SVM은 문서 분류에 있어서 두 가지 장점을 가지고 있다. 첫째, SVM에서 사용될 특징 벡터를 선택하는 과정에서 추출된 단어들을 추가 선택 과정 없이 사용할 수 있다. 둘째, 학습 방법의 인자 값을 선택하는 과정에 있어서 특별한 조율이 필요하지 않는다.

2.2 K-NN(K-Nearest Neighbour)

K-NN[3]는 통계적 접근 방법으로 입력 벡터들의 근접 이웃을 같은 범주로 분류하는 방법이다. 클래스가 알려지지 않은 문서 Di를 분류하기 위해서 K-NN은 K개의 이웃과의 유사도를 계산하여 문서 Di의 클래스를 결정한다.

유사도를 결정하는 방법에는 유클리디언 거리, 절대 차이, 최대 거리, 민코스키 거리와 같은 방법이 있다.

2.3 베이의 분류자(Naive Bayes Classifier)

베이의 분류자[4]는 확률 모델을 기반으로 하는 분류 기법으로, 각 단어들이 서로 독립적이라 가정한다. 또한, 단어가 나타날 확률은 문서 내에서의 위치와도 독립적이라 가정한다.

베이의 분류자의 훈련과정은 어떤 범주 c_k에서 어떤 단어 w_k가 출현할 확률 $P(w_k|c_k)$과 어떤 범주 c_k가 출현할 확률 $P(c_k)$를 학습하는 것으로, 다음과 같이 구해진다.

$$P(c_k|d) = \frac{P(c_k)P(d|c_k)}{P(d)} \quad (2)$$

여기에서, d는 문서 벡터이다.

III. 특징 추출

XML 문서를 클러스터링하기 위해서는 문서 전체의 특징을 추출해 사용해야 한다. 문서의 특징을 결정지을 수 있는 정보가 사용되어야 한다. 특징 추출 방법으로 태그 특징 벡터와 문단 특징 벡터를 이용한다.

3.1 태그 특징 벡터

XML 문서를 클러스터링하기 위해서는 특징 추출이 중요하다. 특징이란 그 문서를 구별할 수 있는 구성요소이어야 한다. XML 문서 클러스터링에서 사용하는 특징은 주로 XML 문서 내에 존재하는 단어이다. 문서 클러스터링을 위해서는 이런 정보 등을 이용해 특징 벡터를 구성한다.

XML 문서의 특징 중 하나가 사용자가 문서의 태그를 정의할 수 있다는 점이다. 문서의 태그는 문서의 내용을 결정할 수 있는 중요한 기능을 수행하는 단어이다. 태그는 XML 문서의 요소(element)의 한 부분이다. 요소가 XML 문서에서는 중요한 구성 인자가 된다[8].

본 논문에서는 XML 태그를 본문의 단어와 구별하여 특징을 추출한다. 같은 XML 태그가 존재하고 있는 문서는 어떤 다른 XML 문서보다도 같은 종류의 XML 문서일 확률이 높을 것이다. 이런 중요한 특징을 XML 문서 클러스터링의 중요한 인자로 사용한다.

태그 특징 벡터 부분은 태그의 유무에 따라 이진값, 즉 wi,j ∈ {0, 1}로 한다. 여기서, wi,j는 문서 Dj에 있는 태그 키워드 Ki의 가중치를 나타낸다.

3.2 문단 특징 벡터

특징 벡터 중 하나인 문단 특징 부분은 문단에서 추출된 단어를 이용한다. 먼저, 전처리를 걸쳐 불용어를 제거하고 스테밍 작업을 수행한다. 그리고 나서, 문단 특징 벡터를 형성하기 위해서 TF×IDF 색인 가중치 방법을 사용한다. TF×IDF는 가장 많이 사용되는 단어 가중치 계산 방법이다.

TF×IDF의 의미는 하나의 문서 내에서 많이 나오면서, 전체 문서 집합에서는 적게 나오는 단어가 그 문서를 특징짓는데, 큰 가중치를 갖는다는 것으로 다음과 같이 구한다.

$$TF \times IDF(D_i) = \left(\frac{fl^i_1}{N_{tw}} \log \frac{M}{df_1}, f \cdots, \frac{f^i_m}{N_i} \log \frac{M}{df_m} \right) \quad (3)$$

IV. 시스템 구성

XML 문서를 코호넨 맵을 이용하여 태그 특징과 문단 특징으로 구별하기 위한 시스템의 전체 알고리즘은 다음 (그림 1)과 같다.

먼저, XML 문서 컬렉션에서 각 XML 문서에 대해서 전처리를 수행한다. 전처리를 수행하기 위해서 단어별로 파싱을 하고 파싱시에 태그와 본문의 단어를

구별한다. 전처리 과정에서는 불용어(stopword) 제거와 스테밍(stemmin)이 이루어진다. 본 논문에서는 포터(Porter)의 스테밍 알고리즘을 사용한다[9]. 다음으로, 태그는 모두 수용하여 이진값 수식을 사용한다. 문단 단어에 대해서는 빈도수를 계산하여 한계값 이상의 단어에 대해서만 TF/IDF를 적용한다. 태그 특징과 단어 특징이 결합된 특징 벡터가 만들어진다. 그리고 나면, 코호넨 학습을 수행한다.

1. XML 문서에 대해서 파싱을 수행하여 태그와 본문의 단어를 구별한다.
2. 본문의 단어들에 대해서 전처리를 수행한다.
3. 태그는 모두 수용하고 단어에 대해서는 한계값 이상의 단어에 대해서만 특징 벡터에 수용한다.
4. 태그에 대해서는 이진값 수식을, 문단 단어에 대해서는 TF/IDF로 특징 벡터을 생성한다.
5. 코호넨 맵의 초기 구조를 결정한다.
6. 연결 강도를 초기화한다.
7. 특징 벡터를 적용, 모든 뉴런 간의 거리를 계산한다.
8. 최소 거리에 있는 출력 뉴런을 선택한다.
9. 선택 뉴런과 그 이웃 뉴런의 연결 강도를 재조정한다.
10. 수행 횟수가 초과하면 멈추고 그렇지 않으면 5부터 9까지 반복하여 수행한다.

그림 1. 시스템 전체 알고리즘
Fig. 1. Algorithm of System

학습을 위한 첫 단계는 네트워크의 구조와 학습에 관여하는 파라미터들을 설정하는 것이다. 먼저, 입력층과 출력층의 크기를 정한다. 입력층과 출력층의 크기가 정해지면 입력층과 출력층은 완전 연결시키고, 연결 강도 벡터의 값들은 0과 1사이의 임의의 수로 초기화된다.

다음으로 학습에 관여하는 파라미터들을 설정한다. 이때 설정해야할 파라미터는 다음과 같다.

$$P = \{\beta_{initial},\ \beta_{final}, H_{initial}, H_{interval}, Q\} \quad (4)$$

여기에서, $\beta_{initial}$은 초기 학습율, β_{final}은 최종 학습율, $H_{initial}$은 초기 이웃 반경의 크기, $H_{interval}$은 이웃 반경의 크기를 줄일 간격, Q는 총 훈련 횟수이다.

훈련이 시작되어 입력 패턴이 들어오면 승자 뉴런은 입력 벡터와 가장 유사한 연결 강도 벡터를 가지는 뉴런이 된다. 입력 벡터와 연결 강도 벡터의 유사도를 측정하기 위한 함수는 다음과 같이 유클리디언 거리를 사용한다.

$$Dist_{row,\,col}(t) = \sqrt{\sum_{k=0}^{k<\|p\|} (n_{o,k}(t) - w_{0,k\to 1,(rows,\,cols)}(t))^2} \quad (5)$$

여기에서, row와 col는 출력층 뉴런의 위치이다. 승자 뉴런이 결정되면, 이웃 반경의 크기에 따라 연결 강도에 참여할 뉴런들을 결정하고, 연결 강도를 조정한다. 교사 학습(Supervised Learning)과 달리 목표 출력값이 없기 때문에 입력 벡터에 보다 유사해지도록 연결강도를 다음과 같이 조정한다.

$$w_{0,k\to 1,(i,j)}(t+1) = w_{0,k\to 1,(i,j)}(t) + \Delta_{0,k\to 1,(i,j)}(t+1)$$

$$\Delta_{0,i\to 1,(i,j)}(0) = 0 \quad (6)$$

$$\Delta_{0,k\to 1,(i,j)}(t+1) = Neigh_{Win_{row}(t+1),\,Win_{col}(t+1)}(t+1)\ \beta(t)(n_{0,i}(t+1) - w_{0,k\to 1,(i,j)}(t))$$

훈련이 반복되면, $H_{interval}$의 값에 따라 이웃 반경의 크기가 변화되고, $\beta_{initial}$과 β_{final}의 값에 따라 학

습율이 변화된다. 학습 정지 조건으로는 정해진 훈련 횟수를 수행하거나, 모든 입력 패턴에 대해 승자 뉴런의 가중치 벡터 w_q와 바로 인접한 뉴런 중 가장 차이가 큰 가중치 벡터 w_f를 가지는 뉴런 사이의 오차가 허용 오차 E내에 들어오면 충분히 클러스터링되었다고 판단할 수 있기 때문에 학습을 종료시킨다.

V. 실험 및 평가

5.1 시스템의 설계 및 구현

[그림 2]는 코호넨 맵을 시스템으로 구현한 그림이다. 먼저, 크기를 결정하고 관련 파라미터를 초기화 한다. 코호넨 맵을 위한 특징 벡터를 입력 데이터로 사용하고 네트워크의 상태를 저장할 출력 파일을 선택한다. 그리고 학습을 시작하면 된다. 네트워크가 학습될 때마다 특정 클러스터링이 결정되면 승자 뉴런이 선정되고, 선정될 때 마다 값을 변화시켜 선정 횟수의 변화를 나타낸다.

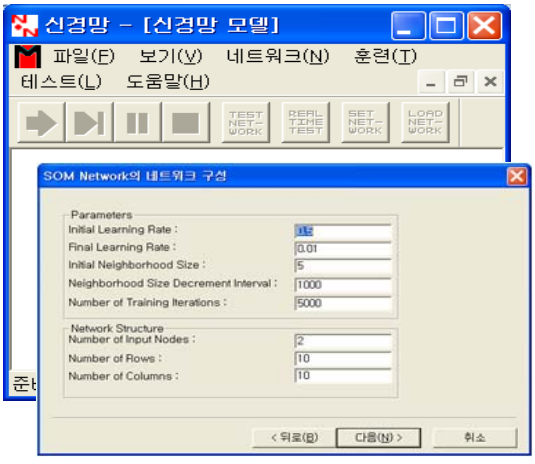

그림 2. 구현된 시스템
Fig. 2. Implemented System

5.2 문서 집합

본 논문에서는 클러스터링 실험을 하기 위해서 문서 집합으로 Reuters-21578 문서 집합을 사용한다[13]. 21578은 문서 집합 내의 문서의 수를 나타낸다. SGML 형태로 만들어져 있고, dtd도 제공된다. 그러므로, SGML의 서브셋인 XML 문서를 테스트하는 데는 적합한 문서 집합으로 여겨진다.

Reuters-21578 문서 집합은 다섯 개의 범주로 구성되어 있다. 각각의 범주는 다시 여러 개의 세부 범주로 분류된다.

5.3 실험용 특징 벡터의 구성

XML 문서 클러스터링을 하기 위해 먼저, 훈련 데이터와 테스트 데이터로 분리한다. 훈련 데이터로 코호넨 맵을 비교사 학습하여 구성하고 테스트 데이터로 정확도를 테스트한다.

본 실험에서는 ModApte 분류를 사용한다. 훈련 집합의 수와 테스트 집합의 문서의 수가 적당히 나누어져있기 때문이다. TOPICS 범주에서 상위 10개의 세부 범주에 속하는 문서들을 실험 데이터로 사용하였다.

먼저, 태그을 분리하고 <BODY></BODY> 사이에 있는 기사 본문을 전처리를 거쳐서, TF×IDF 값을 구하여 문서 벡터를 구성하였다. 차원 축소를 위해서 DF 값이 5 미만인 단어는 제거하고 실험하였다.

5.4 실험 결과

본 논문에서 구성한 특징 벡터를 이용하여 코호넨 맵에 적용하여 실험을 실시하였다. 맵의 크기는 20×20으로 구성하였고, 정지 조건 E는 0.005를 사용하였다.

도출된 결과는 [표 1]와 같다.

표 1. 실험 결과
Table 1. Experimentation Result

세부 범주	재현율(%)	정확율(%)
Earn	48	52
Acq	50	54
Money-fx	44	42
Grain	48	46
Crude	42	44
Trade	50	48
Interest	46	47
Wheat	44	42
Ship	44	40
Corn	43	40

실험 결과에서 재현율과 정확률은 50%를 기준으로 등락을 보였다. 이 실험에서는 Earn과 Acq 범주는 다른 범주에 비해 재현율과 정확율이 조금 좋게 결과가 도출됐다. 이는 문서의 수가 다른 범주에 비해 많았기 때문이라고 여겨진다.

5.5 타 시스템과의 비교 실험

태그와 단어를 분리하여 구성한 특징 벡터와 단지 단어만을 사용하여 특징 벡터를 구성한 경우에 대해서 SVM과 K-NN을 적용한 시스템과 비교하여 실험을 실시하였다. [그림 3]은 실험 결과를 그래프로 나타낸 것이다.

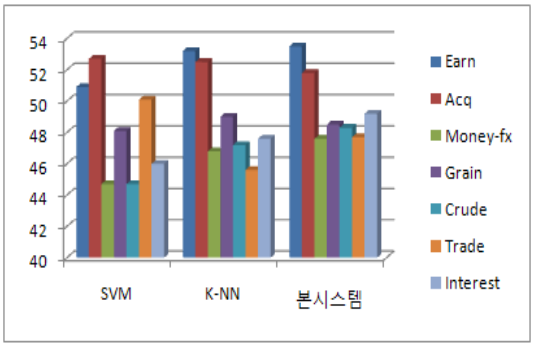

그림 3. 타 시스템과의 비교 분석
Fig. 3 comparative Anaysis with other systems

그림에서 보듯이, 정확율의 경우는 본 시스템이 전체적으로 10%정도 성능이 좋게 나왔다. 본 논문에서 제안한 시스템은 비교사학습 네트워크이다. 대체적으로 교사 학습의 경우에는 범주의 수가 정해져 있기 때문에, 모든 문서가 범주로 분류되어 재현율과 정확율의 성능이 좋게 나올 수 있다. 하지만, 본 논문에서 사용한 방법론에 의하면 비교사학습과 교사 학습간의 차이가 별로 없음을 알 수 있다. 이 점으로 미루어 본 방법론의 성능이 우수함을 알 수 있다.

VI. 결론

본 논문에서는 XML 문서에 대해서 XML 기법의 중요한 특징 중 하나인 임의의 태그 생성 기법을 이용하여 태그와 문단의 단어를 분리하여 특징 벡터를 생성하여 코호넨 맵에 적용하여 클러스터링을 수행하였다. 태그와 단어를 분리한 특징 벡터를 이용하여 코호넨 맵에서 실험을 실시하여 재현률과 정확률을 테스트하였다. 또한, 전통적인 자동 문서 분류 알고리즘인 SVM과 K-NN과 비교 실험도 수행하였다. 실험 결과에서 재현률과 정확률은 50%를 전후에서 나타났다. 카테고리가 정해진 분류 시스템인 SVM과 K_NN 시스템과의 비교에서 전체적으로 10%정도 성능이 좋게 나왔다. 본 방법론의 우수한 성능을 입증해 주는 바라고 여겨진다. 앞으로, 코호넨 맵의 변형인 성장 그리드와 GH-SOM에의 적용도 필요하다고 본다.

본 논문의 방법론은 검색 엔진에서 사용이 가능하다고 본다. 웹에서의 검색 엔진은 문서의 분류에 관심이 많다. 웹에서 XML 문서의 사용이 많아질수록 XML 문서의 분류에 관심이 많이 집중될 것으로 여겨진다. 이 때, 분류 시스템에 사용 가능하다. 또한, 전문가 검색 엔진처럼, 특정 분야의 검색 시스템을 만들 때, 해당 분야의 문서인지의 여부를 결정하는 시스템에도 사용이 가능할 것으로 여겨진다.

참고문헌

[1] C. Cortes and V.N. Vapnik, "*Support Vector Network*", Machine Learning, vol. 20, pp. 1-25, 1995.

[2] F. Sebastiani, "*Machine learning in automated text categorization*", ACM Computing Surveys(CSUR), Vol.34, Issue1, pp.1-47, 2002.

[3] Y.Yang and X. Liu, "*A re-examination of text categorization methods*", Proc. of the 22nd annual international ACM SIGIR Conference on Research and development in Information Retrieval, Berkeley, California, USA, pp 42-49, Aug.15-19, 1999.

[4] A. McCallum and K. Nigam, "*A comparison of event models for Naive Bayes text classification*", Proc. of the 1st AAAI Workshop on Learning for Text Categorization, Madison, WI, US, pp.41-48, 1998.

[5] Teuvo Kohonen, "*Self-organized formation of topologically correct feature maps*", Biologicl Cybernetics, Springer Berlin, Heidelberg, Vol.43, No,1, pp59-69, 1982.

[6] Teuvo Kohonen, "*The Self-Organizing Map*", Proceedings of the IEEE, Vol. 78, No. 9, September 1990.

[7] T. Joachims, "*Text Categorization with Support Vector Machines: Learning with many relevant features*", Proc. of ECML-98, 10th European Conference on Machine Learning, No.1398 in LNCS, Chemnitz, DE, pp.137-142, 1998.

[8] XML 문서 표준, http://www.w3.org/TR/2008/REC-xml-20081126/

[9] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, *Modern Information Retrieval*, ACM press, pp. 131-160, 1999.

[10] Reuters-21578, http://www.daviddlewis.com/resources/testcollections/reuters21578

[11] 김창수, "비즈니스 정보 공유를 위한 XML 기반의 비즈니스 문서 변환 시스템에 관한 연구", *한국지식정보기술학회 논문지*, 제4권 제3호, pp.35-42, 2009.

박사준(Sa-Joon Park)

1990년 중앙대학교 전자계산학과(학사)
1994년 중앙대학교 대학원 컴퓨터공학과 (석사)
2004년 중앙대학교 대학원 컴퓨터공학과 (박사)
2005년~현재 대구한의대학교 모바일콘텐츠학부 조교수
※ 관심분야 : 인공지능, 시맨틱 웹, 모바일 콘텐츠

박현근(Hyun-Gun Park)

1984년 홍익대학교 기계공학과(학사)
1988년 연세대학교 산업교육(석사)
1995년 서강대학교 정보처리학과(석사)
2005년 중앙대학교 컴퓨터공학과(박사)
1989년~현재 숭실대학교 전산원 e비즈니스경영학과 교수
※ 관심분야: 시맨틱 웹, 인공지능, 웹 에이전트