규칙기반 단어 클러스터링에 의한 문서 분류의 성능 향상

현우석°
한국성서대학교 정보과학부
wshyun°@bible.ac.kr

Performance Improvement of Document Classification by Rule-based Word Clustering

Woo-Seok Hyun°
Dept. of Information Science, Korean Bible University

요 약

분류되지 않은 문서의 문서 분류는 현재까지 아주 중요한 문제로 대두되고 있다. 컴퓨터를 이용한 문서 검색 엔진인 Citeseer에서는 문서 인덱싱을 하기 위해서 자동문서 분류 방법을 사용하고 있다. 문서 분류는 원본 문서의 단어들을 제1의 속성 표현으로 사용한다. 그러나 이와 같은 표현은 고차원과 속성 부족을 초래하게 된다. 단어 클러스터링은 속성 차원과 속성 부족을 감소시키기 위한 효율적인 방법이며 문서 분류 성능을 향상시켜 준다. 본 연구에서는 클러스터 속성 표현을 위한 도메인 규칙기반 단어 클러스터링 방법을 사용한다. 클러스터는 다양한 도메인 데이터베이스들과 단어 철자 속성들로부터 생성되는데, 이와 같은 클러스터 속성 표현은 중요한 차원 감소뿐만 아니라 문서 헤더 라인의 평균 분류 성능에서 향상을 보여 주었고, 원본 문서 단어 기반 속성 표현과 비교해 보았을 때 도서목록 항목 추출의 정확도를 향상시켰다.

1. 서론과 기존 연구

자동 문서 메타 데이터(meta data) 추출은 이질적인 디지털 도서관을 위한 통합된 서비스를 구축하기 위해서 시작되었는데, 데이터베이스에서 정교한 질의를 가능하게 하고 시멘틱 웹의 구현을 용이하게 해 준다. 문서 메타 데이터는 문서 헤더(header)와 도서목록 항목으로부터의 메타 데이터를 말하는데 본 연구에서의 문서로는 연구논문을 사용하였다. 문서 헤더 메타 데이터 추출을 위해서 HMM(hidden markov models)을 사용한 연구[1], 도서목록 항목을 위해서 가변길이 출력 hidden markov 모델을 사용한 연구[2] 등 문서 메타 데이터 추출에 대한 여러 가지 연구가 현재까지 수행되고 있다. 이러한 방법들은 속성 표현을 위해서 원본 문서 단어들을 사용한다. 이 방법들은 단어 통계에 근거한 분류자(classifier)를 훈련하기 위한 통계적 방법을 사용한다. 이러한 표현의 단점으로는 고차원과 속성 부족(sparseness)을 초래하게 되며, 이것은 연산 비용을 증가시키며 분류 성능에도 영향을 줄 수 있다.

성공적인 속성 차원 감소 방법인 LSI(Latent Semantic Indexing)[3], 확률 버전인 PLSI(Probabilistic Latent Semantic Indexing)[4] 등은 차원을 감소시키기 위해서 문서와 단어를 연결시킨다. 유용한 속성을 선택하는 속성 선택 방법은 정보 습득, 문서 빈도 등 계산에 근거한다[5]. 단어 클러스터링 방법[9]은 유사한 문법 혹은 의미 범주에서의 단어들과 같이 유사한 단어들을 클러스터하게 되고 문서 분류를 위한 속성을 클러스터 레이블로 사용하게 된다. 단어 클러스터링은 속성 차원(dimensionality)뿐만 아니라 속성 부족(sparseness)도 감소시켜 준다. 또한 단어 클러스터링은 일반 특성을 고려하고 개별 속성에서 특정 특성을 무시함에 의해서 특정 속성을 일반화한다. 대표적인 단어 클러스터링 방법인 분산 단어 클러스터링[6,7,8]은 문서 분류에서 주요한 성능 향상을 보여 주고 LSI와 PLSI의 성능을 능가한다.

본 논문에서는 제목, 저자, 요약 등 문서의 문법적인 구조에 따라서 단어를 클러스터하는 방법을 사용한다. 속성 같은 클러스터를 사용함에 의해서 목표 클래스의 보다 많은 대표 속성을 가지게 되는데 이것들은 추출될 메타데이터와 유사하다.

단어 클러스터링은 전문영역(domain) 데이터베이스와 단어 철자 속성들[8]을 기본으로 하는데, 이것들은 특정 클래스의 연역적 지식을 가지고 있다. 전문영역은 문서 분류 작업에서 클래스에 해당한다. 전문영역 데이터베이스는 저자 클래스를 위한 이름 단어 데이터베이스가 될 수 있다. 특정 단어들은 전문영역 데이터베이스에서 소속함수(membership)에 의해 클러스터된다. 예를 들어 이름 단어 데이터베이스에서 나타나게 되는 "Tom", "Laura" 등의 단어는 클러스터 되어서 ":name word:" 라는 클러스터 레이블로 표현된다. 유사하게 "Maryland"는 ":state"로 표현된다. 속성 표현의 이러한 유형을 클러스터 속성 표현이라 부른다.

단어 철자 속성은 단어와 단어가 지니는 특별한 특성 혹은 숫자의 사건들을 고려한다. 하나의 단어는 일련의 문자들로 이루어진다. "@"은 이메일의 철자 속성이며 특정 이메일 주소를 클러스터하기 위해 ":email"로 사용된다. 일곱 개의 숫자들은 클러스터되어서 ":digit[7]"으로 표현된다. 이와 같은 단어 철자 속성은 선행 문서 처리 과제들에서 효과적으로 사용되어 졌다[10].

본 논문에서 사용한 단어 클러스터링 방법은 계산 비용을 감소시켰고, 문서 헤더라인 분류 성능과 도서목록 항목 추출 성능을 향상시켰다.

2. 연구 배경

단어 클러스터링은 그림 1에서 보는 바와 같이 메타 데이터 추출 전 단계로서 속성을 표현하는 작업의 일부분이다. 원본 문서 단어들이 클러스터되고 향후 문서 처리 전에 클러스터 레

이블로 변형된다.

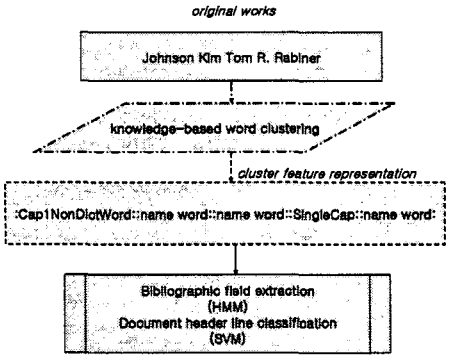

그림 1 단어 클러스터링과 메타 데이터 추출의 순서도

3. 규칙 기반 단어 클러스터링 방법

규칙기반 단어 클러스터링[9]은 다음과 같이 세 단계로 구성되어 있다.

첫 번째 단계는 전문영역 데이터베이스를 구축하는 것이다. 본 논문에서는 외부 전문영역 데이터베이스와 구조화된 전문영역 데이터베이스를 구축한다. 외부 전문영역 데이터베이스는 World Fact Book[11]으로부터 수집되었는데, 1381개의 이름들과 3961개의 성들 그리고 중국 성들이 수집되었다. 이런 자원들은 미국 도시 이름과 다른 나라의 주요 도시 이름(city)의 데이터베이스와 미국 주 이름 데이터베이스와 미국 우편번호 데이터베이스 등으로부터 형성되었다. 이 때 Gazetter[12]도 사용되었다. "pubnum" 등 유용한 외부 데이터베이스가 없는 클래스에 대해서는 문서 빈도(Document Frequency) 측정에 의해서 전문영역 데이터베이스를 구축하였다. 높은 등급의 단어들로는 구조화된 전문영역 데이터베이스를 구축하였다. 표 1은 "affiliation", "note", "pubnum", "phone" 클래스를 위한 높은 등급 단어들을 보여준다. 구조화된 전문영역 데이터베이스는 address 클래스처럼 외부 전문영역 데이터베이스에 있는 클래스라 할지라도 보완적인 정보를 제공해 준다.

표 1 4 가지 클래스에서 문서 빈도에 의한 높은 등급 단어들

Affiliation		Note		Pubnum		Phone	
DF	Feature	DF	Feature	DF	Feature	DF	Feature
313	university	98	research	41	report	18	fax
78	univ	76	support	38	technical	14	tel
126	department	68	grant	21	tr	11	phone
99	institute	54	science	5	csrp	3	usa
57	laboratory	42	part	3	memo		

두 번째 단계는 클러스터를 설계하는 것이다. 전문영역 데이터베이스와 단어 철자 속성에 근거하여 클러스터를 설계한다. 예를 들어서 문자와 숫자가 혼합되어 있는 단어는 ":Digs[3]::Capwords[3]::Digs[2]" 등과 같이 표현한다. 일반적으로 각 전문영역 데이터베이스는 하나의 클러스터에 해당된다.

세 번째 단계는 규칙을 설계하는 것이다. 단어가 다른 전문영역 데이터베이스와 대등(match)하게 하기 위해서 규칙을 설계하고 단어 철자 속성을 확인하여 단어를 적절한 클러스터에 할당한다. 규칙들은 클러스터를 결정하기 위해서 단어의 다양한 속성들을 고려한다. 예를 들면 네임 단어 데이터베이스에 있으면서 대문자로 시작하는 단어는 ":name-word:" 클러스터에 할당된다. 규칙은 하나의 단어가 여러 개의 전문영역 데이터베이스에 속해 있을 경우 다음과 같은 세 가지 방법으로 충돌을 관리한다. 첫째, 다른 전문영역 데이터베이스를 지니는 단어를 대등하게 하기 위해서 특별한 것에서 일반적인 순서를 따른다. 본 논문에서는 헤더라인 분류 실험에서 전문영역 데이터베이스의 우선순위를 Postcode > Abstract > Keyword > Phone > Month > Addr > City > State > Country > Nameword > Word dictionary 로 설계하였다. 하나의 단어가 두 개의 이름 단어 데이터베이스에 나타나게 될 경우 ":dict-word:" 라는 클러스터 대신에 ":name-word" 클러스터에 할당한다. 둘째, 여러 개의 숫자 코드를 사용하는 하나의 단어가 여러 개의 데이터베이스에 속해 있을 경우에는 부호화한다. N은 단어가 속해 있는 데이터베이스의 수를 나타낸다. 예를 들어 문서 헤더에 "degree" 데이터베이스, "pubnum" 데이터베이스, "note" 데이터베이스, "affiliation" 데이터베이스에 모두 속해있는 단어가 있을 경우 위의 4개의 데이터베이스에서 단어의 소속 함수를 나타내기 위해서 네 자리의 이진 코드를 사용한다. 예를 들면 ":1001:"은 "degree", "affiliation" 데이터베이스에는 나타나지만, "pubnum", "note" 데이터베이스에서는 나타나지 않는다는 것을 의미한다. 셋째, 여러 개의 데이터베이스에 속해있는 하나의 단어는 그 자체가 독립적인 클러스터를 형성한다. 이것은 속성의 보편성 초과(over-generalization)를 완화시켜 준다. 그림 2는 "kim" 이라는 단어의 클러스터 할당을 보여준다. "kim" 은 대문자로 시작하고 address 데이터베이스와 이름단어 데이터베이스와 dictionary에 있다. 데이터베이스의 우선순위에 따라서 "address" 클러스터에 할당함을 보여주고 있다.

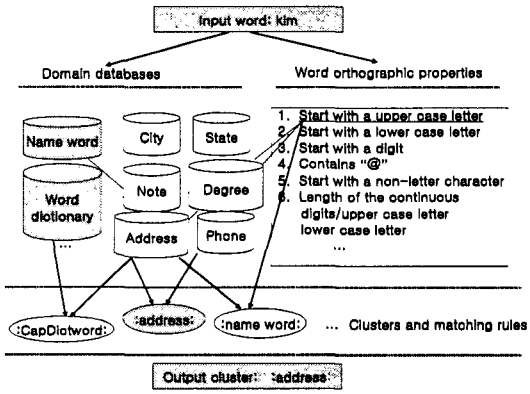

그림 2 단어 "Kim" 에 해당하는 클러스터 할당 예

4. 실험 및 평가

본 논문에서는 문서 헤더라인 분류와 도서목록 항목 추출 두 가지 경우를 실험하였다. 실험 데이터에 대해서 15개의 분류자를 사용하였고, 문서 헤더라인 분류를 위한 방법으로는 SVM(Support Vector machine), 도서 목록 항목 추출을 위한 방법으로는 HMM(Hidden Markov Model)을 사용하였다. 본 논문에서는 규칙기반 단어 클러스터 속성 방법의 세 가지 다른 유형의 속성 표현을 시도해 보았고 각 실험에서 속성 표현의 유형에 대하여 성능을 비교하였다. 훈련을 위해서 450개의 헤더들과 실험을 위해서 415개의 헤더를 이용하였고, 헤더 데이터 집합은 735개의 레이블된 컴퓨터 연구 논문들을 포함하였

다. 도서 목록 데이터 집합은 500개의 레이블된 참고문헌들을 포함하였고, 250개의 훈련 예와 250개의 실험 예로 임의로 나누었다.

문서 헤더라인 분류와 도서 목록 항목 추출의 성능을 계산하기 위하여 두 가지 방법을 사용하였다. 단어 분류 정확도는 다음과 같은 정확도에 의하여, 클래스에 특별한 평가는 다음과 같은 정밀도, 조회율, 정확도와 FMeasure에 의해서 성취되었다.

Precision = A/(A+C)
Recall = A/(A+B)
Accuracy = (A+D)/(A+B+C+D)
FMeasure = (2Precision*Recall)/(Precision+Recall)

단, A는 양성(positive)으로 예견되는 참 양성 예의 수를, B는 음성(negative)으로 예견되는 참 양성 예의 수를, C는 양성으로 예견되는 참 음성 예의 수를 D는 음성으로 예견되는 참 음성 예의 수를 나타낸다.

문서 헤더라인 분류를 위해서 3 가지 다른 속성 표현 유형에 근거하여 7 가지 서로 다른 실험 집합을 비교하였는데 표 2와 같다. 본 논문에서 사용한 클러스터 속성 표현은 특별히 "pubnum", "address", "author", "title" 클래스에서 가장 높은 F Measure를 보여 주었다. 평균적으로 본 논문에서 사용한 규칙기반 단어 클러스터링에서는 원본 단어 표현에 비하여 6.3% 정도의 향상을, 분산단어 클러스터링과 비교했을 때 5.2%의 향상을 보여 주었다.

표 2 서로 다른 속성 표현에 의한 문서 헤더의 라인 분류 성능 비교

Class	1W	1D	2W	2D	3W	3D	4C	Increase	
Title	67.9	71.3	68.1	71.8	72.9	74.8	91.1	10.9	5.7
Author	61.5	69.6	62.2	69.6	62.1	68.9	92.3	26.5	17.6
Affiliation	88.8	892	88.9	88.3	89.2	88.8	90.6	0.7	1.1
Address	81.1	80.2	82.3	80.7	82.1	80.3	92.3	9.9	11.3
Note	69.3	69.0	70.2	67.4	69.0	70.6	64.8	-5.5	-5.7
Email	92.0	51.9	98.7	95.3	98.7	97.7	98.1	-0.5	0.3
Date	83.1	79.2	83.0	79.3	83.3	86.1	93.6	8.4	5.9
Abstract	96.8	97.2	97.0	97.2	95.6	95.5	97.5	0.5	0.3
Phone	63.5	78.9	65.6	77.8	77.1	78.9	78.9	1.8	0
Keyword	65.7	68.9	66.3	68.2	65.1	66.0	69.5	3.1	0.5
Web	96.0	94.1	96.0	96.0	96.0	96.0	96.2	0.2	0.2
Degree	54.6	59.9	56.1	59.9	59.1	57.0	63.2	2.9	1.7
Pubnum	52.6	50.0	52.6	50.0	53.3	50.6	81.6	26.2	29.5

도서목록 항목 추출을 위하여 규칙기반 속성 표현은 원본 단어 표현과 비교했을 때 8.2%의 향상을 보이면서 전체적으로 도서목록 항목 추출 정확도는 89.7%였다. 규칙기만 속성 표현은 속성 차원을 원본 2300개의 단어에서 300개의 클러스터로 감소시켰다. 표 3은 "editor", "page", "tech", "volume" 클래스에서 특별히 전체적으로 클래스에 특별한 추출 성능을 향상시켰음을 보여준다.

5. 결론 및 향후 과제

본 연구에서는 클러스터 속성 표현을 위한 도메인 규칙기반 단어 클러스터링 방법을 사용하였다. 이러한 클러스터 속성 표현은 중요한 차원 감소뿐만 아니라 문서 헤더 라인의 평균 분류 성능에서 향상을 보여 주었고, 원본 문서 단어 기반 속성 표현과 비교해 보았을 때 도서목록 항목 추출의 정확도를 향상시켰다.

향후 연구과제로는 전문영역 데이터베이스 선택을 자동화하여 유용한 단어 철자법을 생성해 내는 향후 연구가 남아있다.

표 3 서로 다른 속성 표현을 사용한 도서목록 항목 단어들의 tagging 성능 비교

Bib field	Original words			Distributional clusters			Our clusters		
	P	R	F	P	R	F	P	R	F
author	95.3	86.1	91.1	87.1	97.7	92.1	96.2	98.7	97.4
book title	92.4	88.5	90.4	81.7	87.3	84.7	88.7	88.7	88.6
date	87.8	82.2	84.7	87.5	82.1	84.5	98.6	95.7	97.1
editor	76.5	45.2	56.9	68.5	60.5	64.5	81.5	63.7	71.1
institution	68.4	78.5	56.9	78.3	71.2	74.6	76.3	77.2	75.8
journal	89.3	65.2	75.4	61.0	63.1	62.3	77.1	78.5	77.9
location	76.5	75.5	76.1	78.8	71.5	75.0	77.8	71.4	74.5
note	58.1	57.4	57.8	32.7	39.4	35.7	76.1	47.3	57.8
pages	71.0	73.5	72.2	66.1	74.0	70.0	95.6	95.7	96.1
publisher	91.3	60.0	68.9	68.7	72.4	70.5	56.0	58.1	57.1
tech	12.2	79.5	21.1	15.7	97.5	27.3	56.3	64.2	59.9
title	87.9	84.0	85.9	96.0	61.8	75.2	92.1	93.0	92.5
volume	85.2	73.2	78.7	81.8	60.5	69.5	87.5	91.6	89.4

참고 문헌

[1] K. Seymore, A. McCallum, and R. Rosenfield, "Learning hidden Markov model structure for information extraction," Proceedings of AAAI99 Workshop on Machine Learning for Information Extraction, 1999.

[2] A. Takasu, "Bibliographic attribute extraction from erroneous references based on a statistical model," Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, pp.49-60, 2003.

[3] S. C. Deerwester, S.T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis", Journal of the American Society of Information Science, Vol. 41, No. 6, pp.391-407, 1990.

[4] T. Hofmann, "Probabilistic latent semantic analysis," Proceedings of Uncertainty in Artificial Intelligence, 1999.

[5] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," Proceedings of the 14th International Conference on Machine Learning, pp.412-420, 1997.

[6] L. D. Baker and A. K. McCallum, "Distributional clustering of words for text classification," Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval, pp.96-103, 1998.

[7] I. Dhillon, S. Manella, and R. Kumar, "A divisive information-theoretic feature clustering for text classification," Machine Learning Research(LMLR), 2002.

[8] N. Slonim and N. Tishby, "The power of word clusters for text classification," Proceedings of the 23rd European Colloquium on Information Retrieval Research, 2001.

[9] H. Han, E. Manavoglu, H. Zha, K. Tsioutsiouliklis, C. L. Giles and X. Zhang, "Rule-based Word Clustering for Document Metadata Extraction," ACM Symposium on Applied Computing, 2005.

[10] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high0performance learning name-finder," Proceedings of ACL Conference on Applied Natural Language Processing, pp.194-201, 1997.

[11]http://www.cia.gov/cia/publications/factbook

[12]http://www.sensus.gov/cgi-bin/gazetter