DOI: 10.3745/KIPSTB.2011.18B.2.093

확장된 벡터 공간 모델을 이용한 한국어 문서 분류 방안

이 상 곤†

요 약

본 논문에서는 한국어 문서의 분류 정밀도 향상을 위해 애매어와 해소어 정보를 이용한 확장된 벡터 공간 모델을 제안하였다. 벡터 공간 모델에 사용된 벡터는 같은 정도의 가중치를 갖는 축이 하나 더 존재하지만, 기존의 방법은 그 축에 아무런 처리가 이루어지지 않았기 때문에 벡터끼리의 비교를 할 때 문제가 발생한다. 같은 가중치를 갖는 축이 되는 단어를 애매어라 정의하고, 단어와 분야 사이의 상호정보량을 계산하여 애매어를 결정하였다. 애매어에 의해 애매성을 해소하는 단어를 해소어라 정의하고, 애매어와 동일한 문서에서 출현하는 단어 중에서 상호정보량을 계산하여 해소어의 세기를 결정하였다. 본 논문에서는 애매어와 해소어를 이용하여 벡터의 차원을 확장하여 문서 분류의 정밀도를 향상시키는 방법을 제안하였다.

키워드 : 벡터 공간 모델, 애매어, 해소어, 전치 인덱스 방법, 상호정보량, 문서분류, 정보검색

Korean Document Classification Using Extended Vector Space Model

Samuel Sangkon Lee†

ABSTRACT

We propose a extended vector space model by using ambiguous words and disambiguous words to improve the result of a Korean document classification method. In this paper we study the precision enhancement of vector space model and we propose a new axis that represents a weight value. Conventional classification methods without the weight value had some problems in vector comparison. We define a word which has same axis of the weight value as ambiguous word after calculating a mutual information value between a term and its classification field. We define a word which is disambiguous with ambiguous meaning as disambiguous word. We decide the strengthness of a disambiguous word among several words which is occurring ambiguous word and a same document. Finally, we proposed a new classification method based on extension of vector dimension with ambiguous and disambiguous words.

Keywords : Vector Space Model, Ambiguous Word, Disambiguous Word, Transposed Index Method, Mutual Information, Document Classification, Information Retrieval

1. 서 론

근래 인터넷의 보급과 함께 전자화된 문서가 범람하고 있다. 대량의 문서에서 사용자가 필요로 하는 문서만을 찾아내기에는 여러 가지 곤란한 문제가 있다. 그러나 문서가 미리 잘 분류되어 있으면 검색의 범위가 좁아져 필요한 문서의 검색 효율이 큰 폭으로 향상하므로 컴퓨터에 의해 자동으로 문서를 분류하는 연구가 꾸준히 필요하다.

분류 기술의 일반적인 방법은 각 분야에 분류되어 있는 문서의 정보를 벡터로 표현한 벡터 공간 모델이 사용된다. 작성된 벡터의 각 축은 단어, 각 축의 가중치는 단어의 출현 빈도값이 사용된다. 실제 문서를 분류할 때에는 새로 입력된 문서에 대해 벡터를 작성하여, 각 분야의 벡터와의 내적에 의해 그 유사도를 계산하고 가장 유사한 분야로 분류된다. 작성된 벡터는 복수 분야의 벡터와 같은 가중치를 갖는 축이 존재한다.

본 논문에서는 여러 분야의 벡터와 같은 가중치를 갖는 축이 되는 단어를 '애매어'라 정의하고, 이를 이용하여 별도의 정보와 함께 벡터를 확장하고, 원래 단어가 가진 애매성을 해소하여 분류의 정확성을 보장하고자 한다. 애매어 모두를 확장하면 그 계산량이 많아지므로 여러 분야를 지칭하는 애매어에 대해서만 벡터를 확장하여 계산량을 감소시킨다. 확장의 대상이 되는 애매어의 추출은 상호정보량[11-15]을 측정하여 그 결과값을 이용하고, 애매성 해소를 위한 정보는 애매어와 함께 동일 문서 내에 출현하는 공기어를 적극 이용한다. 공기어 중에는 한 분야에서만 출현하는 공기어를 '해소어가 될 수 있는 후보어'라 정의하고, 이 해소 후

† 종신회원 : 전주대학교 컴퓨터공학과 부교수
논문접수 : 2010년 11월 18일
수 정 일 : 1차 2011년 1월 18일
심사완료 : 2011년 2월 24일

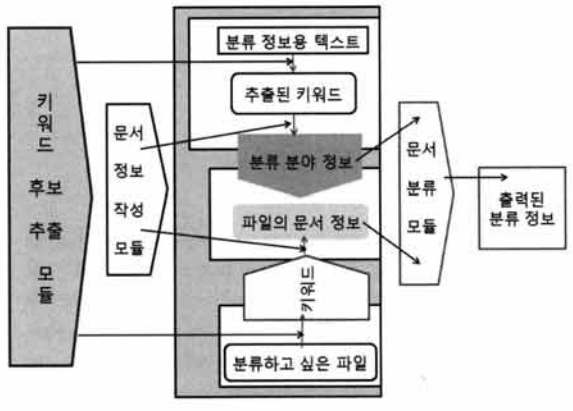
보어를 이용하여 벡터 확장을 한다. 확장 이후의 축은 애매어의 빈도와 공기어와 함께 출현한 애매어의 빈도가 된다. 본 논문에서 제안한 방법의 유용성을 평가하기 위해 언론사에서 제공하는 언론 기사를 이용하여 실험하고 평가한다.

이하 2장에서는 벡터 공간 모델을 이용한 문서 분류 방법, 3장에서는 애매어의 명세 방법, 4장에서는 해소어를 이용한 벡터 차원의 확장에 대해 서술하고, 5장에서는 실험과 평가를 하고, 마지막으로 6장에서 결론을 서술한다.

2. 벡터 공간 모델에 의한 분류 방법

2.1 시스템의 개요

본 연구의 구현을 위해 문서 자동 분류 시스템의 개요를 설명하면 다음과 같다. 문서 자동 분류는 크게 나누어 세 가지의 처리를 포함하고 있다. 키워드 추출 모듈, 문서 정보의 작성 모듈, 그리고 문서 분류 모듈 등 세 가지이다. 아래의 (그림 1)에 전체 시스템의 개요를 나타낸다.



(그림 1) 문서 자동 분류 시스템의 개요

위의 그림에서와 같이 세 가지의 처리 모듈에서 분류 정보를 작성할 때와 새로 입력된 문서를 분류할 때를 따로 나누어 설명한다. 처리의 흐름은 다음과 같다. 먼저 분류 작업에 이용할 분야 정보를 작성하고, 그 위에 새로 입력된 문서에 대한 분류용 문서 데이터를 작성한다. 그것과 이전에 작성된 분야 정보와의 집합을 비교한다. 그 결과를 참조하여 최종 분류를 행하는 순서로 작성한다. 앞에서 언급한 세 가지 처리 중 키워드 추출 모듈과 문서 정보의 작성 모듈은 분류용 문서 데이터를 작성할 때에, 문서 분류 모듈은 문서 정보를 비교하고 분류할 때에 각각 사용된다.

다음에 세 가지 처리에 대해 서술한다. 키워드 추출에서는 문서에서 그 문서의 특징을 가장 잘 설명하는 중요어를 추출한다. 여기서, 중요어와 키워드는 동일한 것으로 간주한다. 키워드 추출은 문헌 검색, 텍스트 편집 등 폭넓은 분야에서 응용되는 기본 기술이다. 현재의 키워드 추출은 크게 나누면 두 가지 방법이 제안되어 있다. 하나는 통제어 방식이고, 다른 하나는 자유어 방식이다. 통제어 방식은 통제어 사전(시소러스)을 사용하는 방식이다. 키워드의 후보어로 가능한 단어를 미리 시소러스 내에 준비하여 두고, 시소러스에 등록된 키워드가 대상 문서 내에 존재하는가에 의해 그 추출 여부가 결정된다[5]. 다른 것은 자유어 방식인데, 이 방법은 시소러스를 사용하지 않고 대상 문서를 형태소 해석하여 그 해석 방법의 기술 수준에 따라 단어를 분할하고 분할된 형태소열에서 키워드 패턴과의 조합이나 빈도 정보 등의 가중치 계산을 통해 키워드를 추출하는 방식이다. 통제어 방식에서 추출 처리는 단순하지만, 관리에 많은 노력이 필요하여 동시에 대량의 시소러스도 필요하다. 반면에 자유어 방식에서는 단어의 추출 처리는 다소 복잡하지만 상대적으로 시소러스의 관리가 필요 없어 통제어 방식에 의한 키워드 추출 방법보다 자유어에 의한 방식이 현재에 많이 사용된다.

문서 정보 작성부는 키워드 추출을 이용하여 추출된 문서의 특징을 나타내는 정보를 서로 비교하기 쉽도록 요약하고 각각의 문서 정보를 작성한다. 문서 정보는 문헌 검색, 문서 분류, 요약 문장 생성 등의 과정에서 이용되는 기본적인 데이터이다. 문헌 검색은 대량의 데이터에서 자신이 지정한 정보를 갖는 데이터를 검색하고 선택하는 방법이다. 문서 분류에서 작성된 문서 정보와 미리 작성되어 있는 분류 체계 정보를 비교하여 그 결과가 가장 좋은 쪽으로 분류해 가는 작업이다. 문서 정보를 저장하는 데이터 방식은 전치 인덱스 방법과 벡터 형식의 표현 방법 등이 있다.

문서 분류 모듈은 문서 정보 작성 모듈에서 작성된 분류를 원하는 정보를 특정한 기준에 의해 선별하여 나누는 기술이다. 실제로 분류할 때는 인간이 미리 작성한 분류 정보를 적극적으로 이용한다. 문서 분류 기술은 전자화 된 데이터의 증가에 따라 그 중요성 또한 증가하고 있고 보다 효율적인 기술이 개발되고 있다.

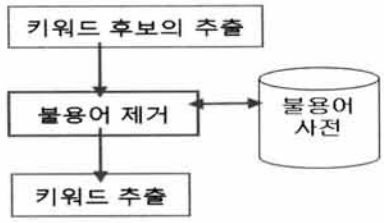
2.2 키워드 추출

2.2.1 명사의 추출

문서 정보인 동시에 분류 정보가 되는 벡터 정보를 작성할 때에는 키워드 추출 방법이 이용되지만, 거기에는 입력된 문서 데이터의 철저한 해석 또한 필요하다. 일반적인 키워드 추출 방법은 형태소 해석이 이용되지만 대량의 데이터를 취급할 때는 형태소를 해석하는데 꽤나 시간이 걸려 키워드 추출의 속도를 크게 저하시킨다. 본 연구에서는 형태소 해석을 이용하지 않고 명사만을 추출하여 이용한다. 추출 명사 중에서 영문자와 숫자를 제거한다. 단, 한글로 된 숫자는 키워드로서 그 가치를 가지므로 그대로 이용한다. 영문자와 숫자는 그 자체로서 중요어가 되는 경우가 적고 키워드성이 낮으므로 중점 키워드에서 제외하였다. 이상과 같은 내용을 토대로 본 논문에서 제안하는 키워드 추출 과정의 처리 흐름을 아래의 (그림 2)에 표시하고 각 조작 과정에서 필요한 처리를 설명하였다.

1) 키워드 후보의 추출 모듈

문서를 스캐닝하여 키워드 후보를 추출[5]한다. 추출된 키



(그림 2) 명사 추출을 이용한 키워드 추출 방법

워드 후보어는 정렬(문자코드 순)되고 문서 중에서 출현한 전체 빈도를 집계한다.

2) 불용어 제거부

불용어 사전을 검색하여 키워드 집합에서 불용어를 포함하는 단어를 제거한다.

3) 출력부

추출된 키워드 집합을 빈도순으로 정렬하여 출력한다.

전체 처리에서 추출의 중심인 키워드 후보어 추출 알고리즘은 다음과 같다.

• 명사 추출을 이용한 키워드 후보어의 추출 알고리즘

키워드 후보 문자열을 'string', 현재의 스캐닝 위치에 있는 문자를 'ch' 라 한다. 명사 후보어로 인식된 문자열을 'keyword_candidate'이라 부른다.

순서-1 {초기화} : ch를 문서의 선두 문자로 설정하고, string을 초기화 한다.

순서-2 {명사의 판단} : 만일 ch가 명사인 경우에는 string에 ch를 추가하고, 순서-4로 이동한다. 그렇지 않으면 순서-3으로 이동한다.

순서-3 {중요어 결정} : string이 2 문자 이상이면 키워드로 추출한다. 그 후 string을 초기화 한다. 순서-4로 이동한다.

순서-4 {ch의 갱신} : ch를 다음 문자로 갱신한다. 만일 ch가 문서의 끝인 경우, string이 비어 있으면 그대로 종료한다. 그렇지 않으면 순서-3을 수행한 후, 종료하고 순서-2로 되돌아간다.

• 추출 예

예문 : 7월 17일 아침에 발생한 지진은 이란 남부를 중심으로 많은 주택 붕괴의 피해를 가져왔다.

순서-1: ch에 '7' 값을 설정하고, string을 초기화 한다.

순서-2: '7'은 명사이므로 string에 '7'을 추가한다.

순서-4: ch를 다음 문자 "월"에 진행하고 순서-1로 되돌아간다.

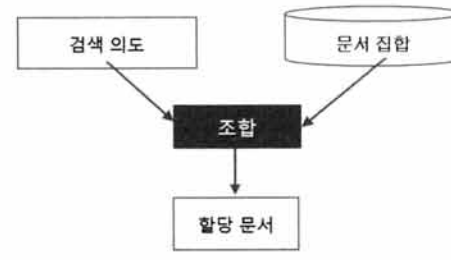
같은 순서를 "1", "7", "일", "아", "침" 으로 반복하여 순서-2는 키워드 후보 문자열이므로 string = "7월 17일 아침"이 추출된다. 동일한 과정으로 "발생", "지진", "이란", "남부", "중심", "주택붕괴", "피해" 가 추출된다.

2.2.2 바이그램

형태소 해석은 앞 절에서 서술한 바와 같이 처리 시간에 문제가 있어 명사 추출을 이용하여 키워드를 추출하고 추출된 키워드에서 바이그램(bigram)을 생성한다[16-17]. 한국어에서 보통명사 대부분이 두 음절로 구성되어 있으므로 2-gram을 이용하였다. 또한 바이그램의 이점은 키워드의 완전 일치 뿐만이 아니라 문자열의 부분 일치도 유사도에 반영되고 평가 시 정밀도가 타 방법에 비해 정확하다. "자연언어"와 "언어처리"의 두 가지 문자열을 예로 들면 완전 일치에는 두 개의 문자열이 전혀 다른 것으로 판단되어 검색 결과에 영향을 주지 않지만, 바이그램이면 "언어"란 2 음절 단어가 일치하여 검색 결과에 반영될 수 있다. 따라서 본 논문에서는 바이그램을 이용하였다.

2.3 문서 정보의 작성

지금까지 문서에 키워드 추출 방법을 적용하여 문서의 특징으로 키워드의 집합을 추출하였다. 이들 키워드들은 그 자체가 문서의 특징을 잘 나타내고 있다. 그러나 이들 모든 키워드들이 문서 분류의 과정에서 직접적으로 사용되지 않는다. 왜냐하면 키워드를 이용하여 두 문서를 서로 비교할 때는 문자열의 비교를 반복해야 하며, 특히 문서의 양이 많아지면 처리 시간이 길어지고 비교하여야 할 데이터의 양도 많아져 여러 어려운 점이 발생한다. 키워드 추출을 이용하여 문서의 특징을 비교하는 방법은 원래의 문서와 별도로 요약된 형태로 문서를 작성하여 따로 보관해야 한다.



(그림 3) 문헌 검색의 개념도

문서 정보는 대량의 문서 데이터를 취급할 때 기본적으로 사용되는 데이터이다. 문서 정보가 사용되는 검색 방법들은 문헌 검색[7], 전문 검색[4], 문서 분류[6] 등이 있다. 이들 방법에 대해 간단히 서술하면 다음과 같다. 문헌 검색은 과거 다수의 서적이나 논문 등 대량의 데이터 집합에서 자기가 지정한 정보를 갖는 데이터를 검색하여 선별하는 방법이다. 문헌 검색의 과정을 도식적으로 표시한 것이 아래의 (그림 3) 이다.

예를 들어, 검색자인 사용자는 「자연언어처리의 방법에 관한 서적이나 논문을 찾고 싶다」라고 생각한다. 이것을 검색 의도라 부른다. 검색의 대상이 되는 것은 과거의 문서들이나 논문과 같이 문서 집합이다. 정보 검색 시스템은 사용자의 검색 의도와 그들의 문서 집합을 조합(照合)하여 그 의도에 적합한 것을 찾아내 처리한다. 여기서 문제가 되는 것은 검색 의도와 문서와의 조합이다. 이들을 원래의 형태 그대로 조합하는 것은 쉽지 않다. 왜냐하면 문서 집합 내에

자연어로 작성된 사용자의 검색 의도를 컴퓨터가 정확히 인식하는 것이 쉬운 일이 아니기 때문이다. 이 문제를 해결하기 위해 쓰이는 것이 '문서 정보' 이다. 검색 의도와 문서를 원래 그대로의 형태로 조합하지 않고, 문서 정보라 하는 중개용 데이터를 부여하여 그들의 조합에 의해 문서를 추출할 수 있게 만들 수 있다.

문서 분류는 작성된 문서 정보와 미리 작성된 분류 정보를 비교하여 그 결과가 좋은 쪽으로 분류 작업을 해 가는 것을 의미한다. 이와 같이 문서 정보를 이용한 문서 분류에 대해 서술하면 다음과 같다. 문서 정보를 저장하는 데이터 방식은 전치 인덱스 방법, 벡터 공간법 등이 있다. 전치 인덱스 방법은 표준적인 데이터 방식이다. 키워드 검색은 이 전치 인덱스 방법으로 실현하는 경우가 많다. 전치 인덱스 방법에서 검색 질문은 통상 색인어와 논리 연산자(∧, ∨, ~)로 구성된다. 이들 논리 연산자는 각각 표준적인 의미를 갖는다. 논리곱 T1∧T2는 두 개의 단어 T1과 T2의 양방향으로 문서에 존재하지 않으면 안 된다는 것을 의미한다. 이에 비해, 논리합 T1∨T2는 두 단어 T1과 T2 중 어느 한쪽에 그 문서가 존재하면 좋다는 것을 요청한다. 부정(~T)은 단어 T가 그 문서에 존재하지 않는다는 것을 요청한다. 예를 들면, 전술의 검색 질의 예문 「자연언어처리의 방법에 관한 서적이나 논문을 찾고 싶다」의 경우를 다시 생각해 보자. 이 예에서 (자연언어처리∧방법)이 검색 시스템에 주는 질의문이다. 또 다른 복잡한 예문을 예시하면 「한국어 이외의 언어에 대한 문맥의존문법이나 문맥자유문법의 학습에 관한 것」을 찾고 싶은 경우는 검색 질의어의 표현은 다음과 같다.

(~한국어) ∧ (문맥의존문법 ∨ 문맥자유문법) ∧ 학습

검색 대상이 되는 문서 집합에는 각각의 문서마다 그 문서의 내용을 대표하는 색인어의 집합이 할당된다. 구체적인 할당 방법은 뒤에서 언급하고 이 절에서는 <표 1>에서 보는 것과 같이 미리 색인어가 할당된 것으로 가정한다. 이 표에서 '1'은 그 문서에 그 색인어가 할당된 것을 나타내고, '0'은 그 문서에 그 색인어가 할당되어 있지 않는 것을 의미한다.

전치 인덱스 방법은 미리 전치 인덱스를 작성하고 이 표를 참조하여 검색 질문과의 조합 연산을 고속으로 수행한다. 여기서 전치 인덱스란 <표 1>의 문서와 색인어를 바꾸어 넣은 것이며, 어느 색인어가 어느 문장에 나타나는가를 의미한다. 이것을 <표 2>에 나타내었다.

〈표 1〉 문서와 색인어

	색인어1	색인어2	색인어3	색인어4
문서1	*1*	*1*	*1*	*0*
문서2	*0*	*1*	*1*	*1*
문서3	*1*	*0*	*1*	*1*
문서4	*0*	*0*	*1*	*1*

〈표 2〉 전치 인덱스의 표현 예

	문서1	문서2	문서3	문서4
색인어1	*1*	*0*	*1*	*0*
색인어2	*1*	*1*	*0*	*0*
색인어3	*1*	*1*	*1*	*1*
색인어4	*0*	*1*	*1*	*1*

이러한 전치 인덱스를 이용하여 예를 들면 「색인어1 ∧ 색인어2」에 합당한 문서를 계산하는 것은 다음의 식과 같이 각각의 색인어에 대해 행벡터의 논리곱을 계산한다.

색인어1 1010 [문서1, 문서3]
색인어2 1100 [문서1, 문서2]

색인어1∧색인어2 1000 [문서1]

보다 복잡한 검색 질문에 대해서도 아래와 같은 과정을 통해 구할 수 있다. 다음은 「(색인어1 ∨ 색인어2) ∨ ~색인어4」에 대한 예를 나타낸다.

색인어1 1010 [문서1, 문서3]
색인어2 1100 [문서1, 문서2]
색인어1∨색인어2 1110 [문서1, 문서2, 문서3]

색인어4 0111 [문서2, 문서3, 문서4]
~색인어4 1000 [문서1]

(색인어1∨색인어2)∧~색인어4 1000 [문서1]

여기에서 주목할 점은 위에서 서술한 방법은 검색 질문에 출현하는 색인어에 대한 행벡터만으로 검색 질문에 적합한 문서를 고속으로 추출할 수 있다. 위의 예에서 색인어의 수와 문서 수가 모두 4 이지만, 실제 검색에서는 어느 쪽이든지 대단히 큰 수가 된다. 따라서 전치 인덱스는 상당히 큰 표가 생성될 수 있지만 검색 질문에 대한 할당 문서를 계산할 때는 그 중 일부만 참조한다.

벡터 형식의 문서 정보에 대해 설명하면 다음과 같다. 벡터 형식의 문서 정보는 검색의 여러 방법 중에 하나인 벡터 공간법과 밀접하게 결합되어 있다. 벡터 공간법은 앞에서 서술한 전치 인덱스 방법의 확장 형태 중 하나로 색인어에 가중치를 부여하여 해당 문서를 순위화 하여 출력하는 방법으로 상당히 고속의 처리가 가능한 검색 방법이다. 벡터 공간법은 문서와 검색 질의어 양쪽 모두를 통일하여 벡터 형식의 문서 정보로 표현하고 이 두 가지 사이에 유사도를 정의하여 유사 문서를 찾는 방법이다.

벡터 형식의 문서 정보는 선형 독립 벡터로 구성되는데, 벡터에 대응하는 키워드에 검색어의 평가값을 곱한 합의 형태로 나타낼 수 있다. 검색어의 평가값은 검색어 사이에서 서로 간의 가중치를 평가할 때 쓰이는 값으로 많이 사용되는 것은 어떤 문서에서 키워드의 출현 빈도나 그 빈도를 모든 출현 키워드 수의 빈도로 나누어 정규화한 값이 있다. t 개의 색인어를 갖는 어떤 문서의 정보는 벡터를 이용하여 이하의 (식 1)과 같이 표현할 수 있다.

$$D = \sum_{i=1}^{t} a_i V_i \qquad \text{(식 1)}$$

여기서, a_i 는 키워드의 색인어 K_i에 대한 값으로 앞에서 제시한 평가값을 사용하는 경우이거나 혹은 단순히 존재 혹은 부재만을 표시하여 존재하면 '1', 부재하면 '0' 이 되는 경우이다. V_i는 그 키워드에 대응하는 벡터이다.

이상과 같이 작성된 벡터를 다른 각도에서 보면 다음과 같다. 단순하게 모든 벡터의 요소가 0 이나 1로 구성되어 있는 경우를 생각하면 그 벡터는 0과 1의 기호열로 구성되어 있다. 새로운 키워드에 대한 벡터를 추가하는 것은 지금까지의 벡터에 대해 새로운 선형 독립 벡터를 추가하는 것과 지금까지 존재하였던 0과 1의 기호열의 한 개의 값을 새롭게 0 에서 1로 정하는 것이다. 따라서 벡터 형식의 문서 정보란 비트열의 각 비트에 키워드를 대응해 가는 매핑 작업과 동일하게 생각할 수 있다.

몇 개의 색인어를 갖는 문서에서 벡터 형식의 문서 정보를 작성하는 과정에 대해 예를 들어 설명하면 다음과 같다. 어느 문서에 대해 키워드를 추출한 결과 {자연언어, 문서분류, 주요어} 라는 키워드가 추출되어 각각의 빈도가 {5, 2, 7}이었다고 하자. 또한 각 키워드에 대응하는 벡터는 각각 {V2, V3, V5}라 하자.

이 때, 이 문서의 벡터 정보는 키워드가 존재하는가 혹은 존재하지 않는가 만을 고려한 경우

$D = 0 \cdot V_1 + 1 \cdot V_2 + 1 \cdot V_3 + 0 \cdot V_4 + 1 \cdot V_5$

이 된다. 이에 비해, 빈도 정보를 고려하면

$D = 0 \cdot V_1 + 5 \cdot V_2 + 2 \cdot V_3 + 0 \cdot V_4 + 7 \cdot V_5$

이 된다.

벡터 정보를 비트열로 간주하면 각각의 벡터를 비트 번호에 따라 대응시켜, 전자의 경우는 작성된 벡터 비트열은 (0 1 1 0 1)이며, 후자의 경우는 (0 5 2 0 7)로 작성할 수 있다.

본 논문에서는 문서 정보를 표현하는 방법으로 벡터 방식을 이용하고자 한다. 그 이유는 벡터 방식에서는 집합론적으로 처리할 수 있다는 이점을 들 수 있으며, 문서 정보 또는 분류 정보를 비트열로 가질 수 있으므로 논리 연산자(AND, OR)로 처리하면 프로그래밍 적으로 용이하며, 비교나 집계 등에 걸리는 처리 시간을 줄일 수 있게 된다는 장점이 있다. 이러한 방법은 문자열을 직접 저장하는 방법, 인덱스 번호를 저장하는 방법에서도 큰 이점이 된다.

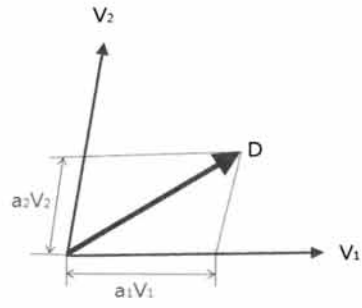
2.4 문서 분류

문서 분류 모듈은 문서 정보 작성 단계에서 작성한 분류하고자 하는 문서 정보를 특정한 기준에 의해 선별하는 기술이다. 분류할 때는 미리 작성된 분류 정보를 이용한다. 인터넷의 많은 정보가 미리 특정한 분류 기준에 따라 정리되어 있으며 개인이 필요한 정보를 즉시 검색하는데 비교적 용이하다. 이와 같이 정보를 분류하는 것은 정보 접근을 지원하는 또 다른 방법이라 생각된다. 이와 같이 문서 분류에 대한 연구는 그 중요성이 증대되고 있다[8].

본 논문에서는 벡터 형식의 문서 정보를 분류 결과에 적극 반영한다. 다음에 그 개념을 설명한다. 전절에서 서술한 바와 같이 어느 문서 데이터의 벡터 정보는 앞서 서술한 (식 2)로 표시할 수 있다. 분류된 문서 정보에 대해서는 (식 3)과 동일한 벡터 정보를 이용할 수 있다. 문서의 벡터 정보는 아래의 식으로 표시할 수 있다.

$$D = \sum_{i=1}^{t} a_i V_i \qquad \text{(식 2)}$$

$$Q = \sum_{i=1}^{t} q_i V_i \qquad \text{(식 3)}$$



(그림 4) 문서의 벡터 표현

위의 식에서 계수 q_i 는 앞의 식 a 와 같은 것이다. 여기서 가장 단순한 경우에 검색 질문에 대해 색인어 T_i 가 존재하는 경우는 1, 존재하지 않는 경우는 0 이 된다. 보다 복잡한 경우는 분류된 문서 Q 에 대해 색인어 T_i 의 중요도는 표시하는 값이 입력된다. 예를 들면, 출현 빈도나 그것을 전체 빈도로 나눈 정규화 된 값이다. (그림 4)에 2차원 벡터 공간에서의 문서 표현의 예를 나타낸다.

이와 같이 본 연구에서는 문서를 벡터의 선형 결합을 이용하여 그 의미를 표현하였으며, 이에 의해 문서 사이의 비교나 분류 작업이 벡터 연산으로 가능함을 입증한다. 벡터 공간에서 두 가지 벡터의 유사도는 여러 형태로 정의되지만 본 연구는 두 가지 벡터가 이루는 각의 코사인 값을 이용한다.

〈표 3〉 동일한 가중치를 갖는 벡터

	단어1	단어2
문서1	10	10
[분야1]	10	5
[분야2]	10	15

〈표 4〉 상호정보량의 계산

	단어 1	단어 2
분야 1	10	5
분야 2	10	15

$$x \cdot y = |x|\ |y| \cos \alpha \quad \cdots\cdots\cdots\cdots\cdots \text{(식 4)}$$

여기서 $|x|$는 벡터의 길이를 나타내고, α는 벡터가 이루는 각을 나타낸다. 이러한 유사도를 이용하는 경우 문서 정보 D 와 질의어 Q의 유사도는 이하의 식과 같다.

$$sim(D, Q) = D \cdot Q = \sum_{i,j=1}^{t} a_i Q_j V_i \cdot V_j \qquad \text{(식 5)}$$

여기서 간단히 하기 위해 t 개의 키워드에 대응한 벡터 V 는 각각 직교한다고 가정한다. 이 때,

$V_i \cdot V_j = \begin{cases} 0, & i \neq j \text{일 때} \\ 1, & i = j \text{일 때} \end{cases}$ 이다.

따라서 $sim(D, Q)$는 다음 식에 의해 간략화 된다.

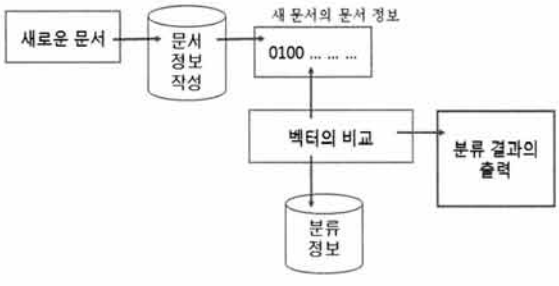
$$sim(D, Q) = \sum_{i=1}^{t} a_i q_i \qquad \text{(식 6)}$$

그러나 내적은 벡터의 크기에 의해 영향을 받으므로 두 가지 벡터가 이루는 각의 코사인 값을 이용하여 이 내적에 기초한 유사도를 정규화 한다. 즉, 내적이 아니고 두 개의 벡터가 이루는 각의 코사인 값을 취한다. 정규화 한 유사도를 $sim'(D, Q)$라 하면 이하의 식과 같이 표시할 수 있다.

$$sim'(D, Q) = \frac{sim(D, Q)}{|D||Q|} \qquad \text{(식 7)}$$

이와 같이 문서 분류 연구에서 벡터를 사용하면 분류 정보와 문서 정보를 비교하는 유사도를 정의할 수 있다. 또한 코사인 유사도는 문서 데이터의 벡터 Q와 분류 정보의 벡터 D 가 이루는 각이 적을수록 두 가지 정보가 유사하게 된다.

앞에서 서술한 두 종류의 문서 분류 방법에서는 벡터 정보를 어떻게 활용하여 문서를 분류할 것인가에 대해 서술하였다. 분류 기술의 개념도를 아래의 (그림 5)에 제시하였다.



(그림 5) 문서 분류기의 개념도

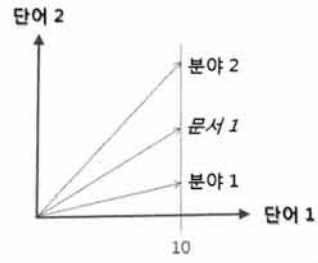
문서 분류에서 사용되는 방법은 인간이 미리 분류하여 작성한 데이터를 기본으로 분류 데이터를 작성하고, 새로 입력된 문서는 그 문서 정보를 작성하여 분류 데이터와의 유사도를 계산한다. 그 값이 높을수록 해당 분야에 가깝다고 할 수 있다.

3. 애매어의 결정

3.1 애매어의 정의

벡터 공간 모델에 의한 문서 분류에서 분류 정보, 문서 데이터는 단어 가중치와 빈도값을 축으로 하는 벡터로 표현된다. 이를 벡터 간 유사도를 계산하여 문서 분류가 이루어진다. 그러나 분류 정보인 벡터의 축에 가중치가 같으면 불리한 경우가 발생한다. 예를 들어 <표 3>과 같은 경우를 생각해 보자.

위의 <표 3>은 분류 정보로 [분야 1], [분야 2]의 벡터가 준비되어 있고, 대상 문서로 문서 1이 입력되어 있다고 가정한다. 각 벡터를 벡터 공간에서 표현하면 (그림 6)과 같다. 문서 1과 각 분야의 유사도를 계산하면 다음과 같다.



(그림 6) 동일 가중치의 축을 갖는 벡터

$sim(\text{문서 1}, [\text{분야 1}]) = 0.949$
$sim(\text{문서 1}, [\text{분야 2}]) = 0.981$

이 결과에서 보면 문서 1은 유사도의 값이 상대적으로 큰 [분야 2]에 관한 문서이고, [분야 2]로 분류될 것이다. 실제로 문서 1이 [분야 2]에 관한 내용이 존재하면 아무 문제가 없으나, 만약 [분야 1]에 관한 내용이 기술되어 있다면 분류의 결과가 잘못된 것이다. 이 때 [분야 1]과 [분야 2]에서 단어 1에 주목하면 모두 단어 1의 가중치가 같게 되고, 결국 단어 1은 [분야 1]과 [분야 2]의 사이에서 '애매한 정보'라 할 수 있다. 이와 같이 이러한 결과에 대하여 아무런 처리도 하지 않은 채로 벡터의 비교를 실행하면 잘못된 결과가 출력될 수 있다.

본 논문에서 예시한 단어 1과 같이 복수[6]의 분야(이 경우에 [분야 1]과 [분야 2])에서 같은 정도의 정보를 갖는 단어를 '애매어'로 정의한다. 이 애매어에 대해 다른 정보를 이용한 벡터를 확장하여 애매성을 해소하고 문서 분류 기술을 향상시킨다. 다음 절에 애매어의 결정 방법에 대해 보다 자세하게 서술한다.

3.2 애매어의 결정 순서

앞에서 애매어를 정의하고, 분야 벡터의 축으로 사용되는 단어가 많을수록 애매어가 출현하는 비율이 많아진다. 이에 의해 애매어의 특정 벡터를 확장하는데 걸리는 시간과 애매어 매칭의 계산량이 많아진다. 여기서 분류된 각 분야를 추출하기 위한 단어로서 애매어를 결정하고, 특정 애매어에 대해서만 벡터를 확장하면 처리하여야 할 계산량을 크게 줄일 수 있다. 애매어의 결정에 대한 순서는 다음과 같다.

- 결정의 순서와 알고리즘

[순서 1] 분류되어 있는 분야와 출현하는 단어 사이의 상호정보량을 계산한다.

[순서 2] 각 분야에서 상호정보량이 높은 단어는 각 분야에 출현하는 단어 수의 α%를 특정 분야로 한정하는 단어라 간주하여 '애매어 후보어'로 추출한다.

[순서 3] 이 애매어 후보 중에서 복수의 분야로 출현하고 있는 것을 최종적으로 '애매어'로 추출한다.

벡터 확장을 하는 애매어는 복수의 분야를 한정[6]하고 있는 것으로 간주한다. 분야를 한정하는 단어를 찾아내는 방법은 처음에는 사람이 선택하는 방법을 생각할 수 있다.

그러나 문서 분류는 대량의 문서를 취급하고, 사람의 많은 노력이 필요하고 개인별(혹은 작업자별)로 선택한 단어에 그 차이점이 크게 발생한다. 분야와 단어의 관련 정도를 나타내는 척도로서 상호정보량을 이용할 수 있다[7]. 분야 C와 단어 T가 나타날 확률이 P(C), P(T) 이며, 분야 C와 단어 T가 공기하여 출현할 확률이 P(C, T)라 하면, 분야 C와 단어 T의 상호정보량 MI(C, T)는 다음의 식과 같이 정의된다.

$$MI(C, T) = \log \frac{P(C, T)}{P(C)\ P(T)} \qquad \text{(식 8)}$$

- $P(C)$: $\frac{\text{분야 } C\text{에 출현한 단어의 총빈도}}{\text{출현 단어의 총 빈도}}$
- $P(T)$: $\frac{\text{단어 } T\text{의 출현 빈도}}{\text{출현 단어의 총 빈도}}$
- $P(C, T)$: $\frac{\text{분야 } C\text{에 출현한 단어 } T\text{의 빈도}}{\text{출현 단어의 총 빈도}}$

이와 같은 상호정보량의 값은 분야 C 또는 단어 T의 연관성에 의해 다음과 같이 세 가지 특징을 갖는다.

첫째, 분야 C와 단어 T가 올바른 상관 관계를 나타내면 P(C, T) > P(C) P(T) 가 되어 결과적으로 MI(C, T) > 0 이 된다.

둘째, 분야 C와 단어 T 사이에 의미 있는 관계가 없는 경우에는 P(C, T) ≒ P(C) P(T) 가 되어 결과적으로 MI(C, T) ≒ 0 이 된다.

셋째, 분야 C와 단어 T가 서로 배반의 관계에 있고, 음의 상관관계를 나타내는 경우에는 P(C, T) < P(C) P(T) 가 되어 결과적으로 MI(C, T) < 0 이 된다.

이와 같이 상호정보량은 단어 T의 출현 빈도가 있는 분야와 기타 분야 사이에서 편차가 있을 때에 큰 값을 갖는다. 따라서 상호정보량이 높은 단어는 그 분야와의 연관성이 강하고, 그 분야를 특징화하는 단어라 생각된다. <표 4>는 어떤 벡터가 주어졌을 때 상호정보량 계산의 실행 예를 나타낸다.

- 출현 단어의 총 빈도 = 40,
- 단어 1의 출현 빈도 = 20, 단어 2의 출현 빈도 = 20,
- 분야 1에 출현한 단어의 총 빈도 = 15,
- 분야 2에 출현한 단어의 총 빈도 = 15

이 때 상호정보량 MI([분야 1], 단어 1)를 계산한다.

- $P(\text{단어 } 1) = \frac{20}{40} = 0.5$
- $P([\text{분야 } 1]) = \frac{15}{40} = 0.375$
- $P([\text{분야 } 1],\ \text{단어 } 1) = \frac{10}{40} = 0.25$

따라서

$$MI([\text{분야 } 1], \text{단어 } 1) = \log \frac{0.25}{0.5 \times 0.375}$$

$$= 0.415$$

동일하게 계산하면

$MI([\text{분야 } 1], \text{단어 } 2) = -0.568$
$MI([\text{분야 } 2], \text{단어 } 1) = -0.322$
$MI([\text{분야 } 2], \text{단어 } 2) = 0.263$

과 같은 결과를 얻는다.

다음으로 상호정보량의 값이 높은 단어가 상위의 몇 번째까지의 단어들이 특정 분야를 한정하는가를 판단하여 이러한 단어군(집합)을 애매어의 후보로 결정할 것인가 하는 문제에 대한 결정이 필요하다. 이것을 해결하기 위한 대책으로 임계값(α)를 설정하였다. 각 분야에서 출현하는 단어의 개수는 분야별로 다르게 출현하므로 단순히 「애매어의 후보를 결정할 때 상호정보량이 높은 단어에서 상위 α번째까지로 한다」라고 일방적으로 정하면 출현하는 단어가 적은 분야에서는 대부분의 단어가 모두 애매어로 할당되어 버리는 문제가 발생한다. 따라서 각 분야에서 애매어 후보가 되는 단어의 개수를 같은 비율로 결정해 줄 필요가 있다. 여기서 임계값 α는 각 분야에서 상호정보량이 높은 단어 중에서 분야에 출현하는 단어 수의 α번째까지의 단어를 애매어 후보로 추출한다.

각 분야에서 애매어 후보로 추출된 단어를 선택하여 벡터 확장을 하고 이를 애매어로 하는 특징을 갖는 단어로 이용한다. 이 애매어들은 복수의 분야에서 애매어 후보로 출현하는 단어 중에서 선택한다. 아래의 (그림 7)은 애매어로 선정된 예를 나타낸다.

(그림 7) 애매어로 선정된 단어의 예

(그림 7)에서 축구, 농구, 야구 등 각각의 분야로 각 단어의 상호정보량을 계산한다. 그 값이 높은 순으로 정렬한다. 그림 속의 가로선이 임계값을 나타내는 위치이고, 그 선 보다 위에 있는 단어들이 애매어의 후보가 된다. 결국 농구의 분야에서 애매어의 후보가 되는 단어는 "프리드로우", "리그", "슛" 등이다. 이들 애매어 후보어들 중 애매어의 특징을 살펴본다.

[축구] 분야에 출현하는 "업사이드"의 경우, "업사이드"는 축구에서만 출현하므로 애매어로 선택되지 않는다. 동일하게, "프리드로우", "홈런"의 경우와 같이 오직 한 분야에서만 애매어 후보로 출현하지 않는 단어는 애매어로 선택되지 않는다. 이 예에서 애매어로 추출되는 단어는 [축구]와 [농

구]의 분야에서 애매어 후보로 선택된 "슛", [축구]와 [야구]의 분야에서 애매어의 후보로 선택된 "리그"이다. 각 애매어의 후보는 여러 분야를 지칭할 수 있는 단어를 애매성을 갖는 단어 즉 애매어로 선정한다.

4. 해소어를 이용한 벡터의 차원 확장

4.1 해소어의 정의

본 절에서는 애매어를 갖는 분야가 가진 문서의 애매성을 적당한 다른 정보를 이용하여 그 애매성을 해소하여 문서 분류의 정확도 향상을 시도하고자 한다. 애매성을 갖는 정보에서 애매성을 해소하는 방법에는 단어가 출현하는 위치를 고려한 방법, 문서에 부여된 태그 정보를 고려한 방법, 동일 문장 또는 문서에 출현하는 단어의 정보를 이용하는 방법 등을 생각해 볼 수 있다.

본 논문에서는 애매성을 해소하는 정보로서 애매어와 함께 동일 문서 중에 출현하는 단어(여기에서는 '공기어'라 부른다)를 이용한다. 본 논문에서 사용한 애매어는 복수 분야 사이에서 애매성을 갖고 있으므로 공기어 중에서 오직 한 개의 분야에서만 출현하는 공기어를 '해소어'라 정의한다.

4.2 해소어의 결정

4.2.1 상호정보량을 이용한 해소어의 결정

애매어가 갖는 애매성을 해소하기 위해 해소어를 결정한다. 해소어의 결정 방법에 관한 알고리즘은 다음과 같다.

*** 해소어 결정의 순서**

[순서-1] 애매한 분야에서 애매어를 포함한 문서를 추출한다.

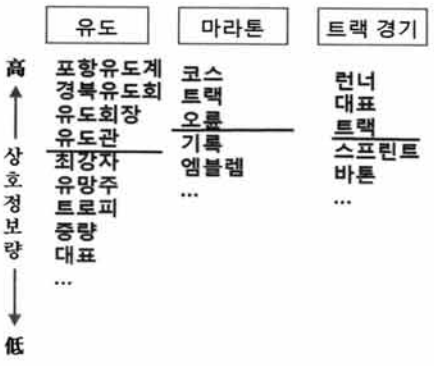
[순서-2] 추출된 문서에 포함된 단어와 애매어가 애매성을 갖고 있는 분야 사이의 상호정보량을 계산한다.

[순서-3] 각 분야에서 상호정보량이 많은 단어에서 출현 단어의 β% 개를 해소어 후보로 추출한다.

[순서-4] 한 분야에서만 해소어 후보로 존재하고, 기타의 다른 분야에서 상호정보량 $MI \leq = 0$ 이 되는 해소어 후보를 추출한다.

[순서-5] 앞의 [순서-4]에서 추출한 해소어 후보 중에서 가장 상호정보량이 높은 것을 '해소어'라 한다.

문서 분류에서 입력은 문서이므로 해소어는 애매어와 동일 문서 내에 공기하고 있는 단어 중에서 결정하여야 한다. 먼저, 애매성을 갖고 있는 분야에서 애매어를 포함하는 문서를 추출한다. 애매어와 공기하여 출현하는 단어는 그 분야에서 쉽게 추출할 수 있는 단어라 할 수 있으므로 애매어를 추출할 때와 같이 단어와 분야 간의 상호정보량을 계산한다. 임계값 β에 의해 각 분야에서 상호정보량이 높은 단어에서 분야에 출현하는 단어 수의 β%번째까지의 단어를 해소어 후보로 추출한다. 추출된 해소어 후보 중에서 해소어를 추출하지만 복수 개의 분야에서 해소어 후보로 출현된 단어를 선택하면 애매성이 해소되지 않는 분야가 출력될 수 있다. 요약하면, 한 분야에서만 해소어 후보로 출현하고, 다른 분야에서는 상호정보량의 값이 0 이하가 되는 단어를 해소어로 추출한다.



(그림 8) 해소어로 선정된 단어의 예

(그림 8)에 애매어 "메달리스트"가 [유도], [마라톤], [트랙 경기]의 세 분야로 애매성을 갖고 있을 때의 해소어 추출의 예를 나타낸다. 이 그림은 [유도], [마라톤], [트랙 경기]의 각 분야에서 애매어 "메달리스트"를 포함한 문서를 추출하여 각 분야와 단어 사이의 상호정보량을 계산하고 그 값이 높은 단어 순으로 정렬한 그림이다. 각 분야에서 사용된 단어는 그 분야와의 상호정보량 값이 올바르다고 가정하고, 쓰이지 않는 단어는 상호정보량이 '0' 이하라 한다. 또한 그림 중의 가로선(—)이 임계값 β를 표시하는 위치이며, 선보다 위에 있는 단어가 각 분야의 출현 단어 수의 β%개 이다.

〈표 5〉 애매어를 포함한 문서에서 해소어와 함께 출현하는 문서 수

	코스	트랙	대표	기록	오륜	판정	런너
[유 도]	0	60	100	30	10	20	0
[마 라 톤]	30	5	0	0	30	0	25
[트랙경기]	20	10	0	0	30	10	40

[마라톤], [트랙 경기]에서 해소어의 후보로 나오는 "트랙"은 "메달리스트"와 함께 출현하여, [유도]가 아니고, [마라톤] 혹은 [트랙 경기]라 판단되며, 두 분야에서 애매성을 갖고 있으므로 해소어로는 선택되지 않는다. 다음으로 [트랙 경기]에서 해소어로 나타나는 "대표"에 대해 살펴보면 확실히 한 개의 분야에서 해소어 후보로 나타나지만, [유도] 분야에서 올바른 상호정보량을 갖고 있다. 이 때문에 애매성을 해소할 수 있다고 말할 수 없고, 역시 해소어로 선택되지도 않는다. 이 예에서 해소어로서의 조건을 만족하고 있는 단어는 해소어 후보로 나타나는 단어에서 앞의 "트랙"과 "대표"를 제외한 것이다. 이 중에서 상호정보량이 가장 높은 단어를 해소어로 결정한다. 이에 반해 [마라톤]의 "코스"가 가장 높았다면 애매어 "메달리스트"에 대한 해소어는 "코스"가 된다. 이에 의해 입력 문서에 "메달리스트"와 "코스"가 함께 출현할 때 입력 문서와 [마라톤]의 유사도가 높게 되도록 반영된다.

4.2.2 공기의 세기를 고려한 해소어의 결정

상호정보량만을 이용하여 해소어를 결정할 때에 한 개의

분야에서는 상호정보량이 높고, 다른 분야에서는 상호정보량이 '0' 이하의 값을 표시하는 단어를 추출한다. 그러나 상호정보량이 '0' 보다 작은 값을 갖는 단어라도 어느 정도의 빈도수를 갖고 있다면 애매어와 함께 출현할 수 있다. 이 때문에 애매어와 해소어의 쌍이 출현하지 않는 분야에서도 출현하는 경우가 있다. 앞의 (그림 8)의 예를 살펴보면 애매어 "메달리스트"와 해소어 "코스"가 함께 출현하는 분야는 [마라톤] 이므로 바람직하지만, [유도]나 [트랙 경기]에서도 출현하지 않는 문서가 존재하고 입력 문서가 [유도]에 대해 쓰인 문서라 하여도 [마라톤] 분야와의 유사도가 강해질 가능성이 있다.

상호정보량을 이용한 공기의 세기를 고려한 애매어가 애매성을 갖는 분야 중 단지 한 개의 분야와 공기하여 출현하고 다른 분야에서는 공기하여 나타나지 않는 단어를 해소어로 결정한다. 이 때 공기의 세기를 측정하는 것으로 임계값 γ를 설정한다. 임계값 γ는 한 개의 분야에서만 공기하여 출현하는 단어가 그 분야에서 애매어를 포함한 문서 수의 γ% 이상으로 출현하지 않으면 안 되는 조건이 부여된다. 이하에 공기의 세기를 고려한 해소어 결정 순서를 나타낸다.

• 공기의 세기를 고려한 해소어의 추출 방법

[순서 1] 애매어가 애매한 분야 중에서 애매어를 포함하는 문서를 추출한다.

[순서 2] 얻어진 문서에 포함된 단어와 애매어가 애매성을 갖는 분야 사이의 상호정보량을 계산한다.

[순서 3] 각 분야의 상호정보량이 많은 단어에서 출현 단어의 β%개를 해소어 후보로 추출한다.

[순서 4] 한 개의 분야만으로 해소어 후보로 존재하고, 기타 분야에서는 상호정보량이 $MI \leq = 0$ 이 되는 해소어 후보를 추출한다.

[순서 5] 해소어 후보로 열거된 분야의 [순서 1]에서 추출한 문서 중 해소어 후보가 γ% 이상의 문서에 출현하고 있는 것을 추출한다.

[순서 6] [순서 5]에서 추출한 해소어 후보어 중에서 상호정보량이 가장 높은 단어를 해소어로 추출한다.

앞의 (그림 7)의 예에서 추출된 해소어 후보에서 공기의 세기를 고려한 해소어의 결정 순서를 이용한 해소어의 특징예를 나타낸다.

<표 5>와 같이 각 분야에서 해소어 후보가 애매어 "메달리스트"와 같이 출현한 문서 수가 결정된 경우를 생각해 보자. 각 분야에서 애매어 "매달리스트"를 포함하는 문서 수는 [유도], [마라톤], [트랙 경기]에서 각각 100, 30, 50이었다고 하면 상호정보량이 높은 순으로 표의 좌측부터 정렬한다.

상호정보량이 가장 높은 "코스"에 대해 살펴보면 [마라톤]의 분야는 언제나 애매어와 함께 출현하지만, [트랙 경기]에서는 20 문서에서 애매어와 함께 출현하고 있다. 해소어로 "코스"를 선택한 경우에 입력 문서에 "메달리스트", "코스"가 함께 출현하였을 때는 [마라톤] 분야와의 유사도를 높일 수 있다. 그러나 이 쌍은 [트랙 경기] 분야에서도 출현할 가능성이 있고, 이때에는 [트랙 경기] 문서임에도 불구하고 [마라톤]과의 유사도를 높이므로 "코스"는 해소어로 선택되지 않는다. 다음으로 "트랙"에 대해 살펴보면, 애매어와 함께 출현하는 문서 수가 60 밖에 없지만 다른 분야에서도 출현하고 있으므로 해소어로써는 부적절하다. "대표"의 경우에는 애매어와 함께 출현하고 다른 분야에서는 절대로 함께 출현하지 않고 있다. 결국에는 "메달리스트"와 "유도관"은 [유도]에서만 출현하지 않아 해소어로 유효하며, 이 경우에는 "유도관"이 해소어로 선택된다.

〈표 6〉 애매어가 존재하는 벡터의 예

	단어1	단어2
[분야 1]	10	20
[분야 2]	10	5

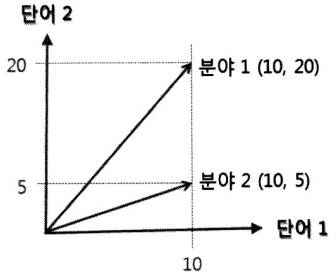
〈표 7〉 차원 확장 후의 벡터

	단어1	단어2	(단어1, 단어2)
[분야 1]	10	20	10
[분야 2]	10	5	0

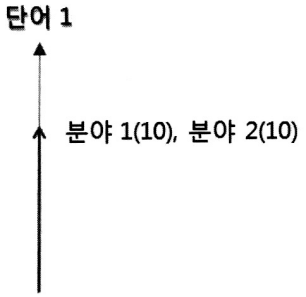
4.3 벡터의 차원 확장

벡터 공간 모델에서 분류 정보는 문서 데이터의 가로축과 단어의 세로축으로 하고, 가중치를 빈도 정보로 한 벡터로 표현되어 그 차원 수는 문서에 출현하는 단어와 다른 단어 수와 같다. 애매어가 갖는 애매성을 해소어 정보를 이용하여 애매성을 해소하기 위해 벡터의 차원을 확장한다. 차원을 확장하기 전 벡터에서 애매어가 축으로 된 부분에서는 애매어가 빈도 값만 존재한다는 정보만 보유하지 않으므로 타 분야와의 애매성이 생겼다. 애매어가 축으로 되는 부분에 애매어와 해소어가 함께 출현하는 축을 새로 작성하여 벡터 차원을 증가시켜 확장한다. 벡터 차원을 확장하는 벡터는 애매어가 애매성을 갖고 있는 분야 벡터 뿐 아니라 모든 분야 벡터 또는 입력 벡터에 대해 수행한다. 왜냐하면 일부 벡터에 대하여 벡터의 차원을 확장하면 유사도 계산 시에 차원수가 다른 벡터와의 비교가 이루어지기 때문이다. 새로 작성한 축의 가중치는 애매어와 해소어가 함께 출현하는 빈도값이다.

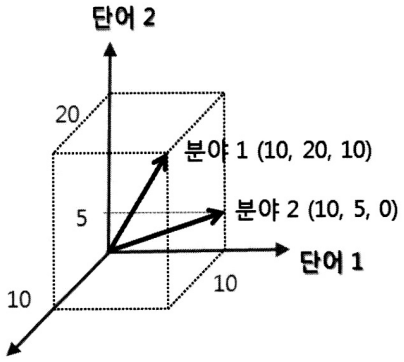
<표 6>은 애매어를 포함하는 벡터가 주어진 경우에서의 벡터 차원의 확장 예를 표시한다. "단어1"을 애매어, "단어2"를 해소어라 하고, 입력 문서에 "단어1"과 "단어2"가 함께 출현할 때는 [분야1]에 대한 유사도가 높아진다. <표 6>에 표시된 각 벡터를 벡터 공간에 표현하면 (그림 9)와 같으며, 애매성을 갖고 있는 축을 (그림 10)에 나타내었다. (그림 10)에서 명백하게 애매어가 축이 되는 부분에서는 복수의 분야 벡터가 같은 벡터로 표현됨을 알 수 있다. <표 6>에 표시한 벡터에 (애매어, 해소어) 축을 (단어1, 단어2)를 작성하여 벡터 차원으로 확장한다. 확장한 벡터를 앞의 <표 7>에 표시한다.



(그림 9) 벡터 공간에서의 표현

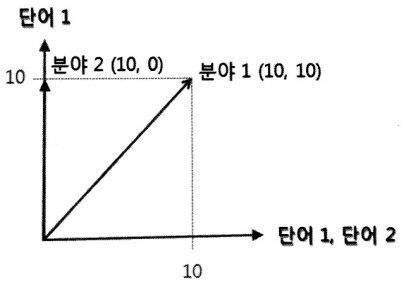


(그림 10) 애매성을 갖는 축



(그림 11) 벡터 공간에서의 표현

<표 6>을 벡터 공간으로 표현한 것을 (그림 11)에 표시한다. 애매어와 차원 확장한 축에 대해 살펴 본 예를 (그림 12)에 표시한다. (그림 9)와 (그림 10)을 비교하여 보면 벡터들이 전혀 다름을 알 수 있다.



(그림 12) 차원을 확장한 벡터

4.4 유사도 계산

분류 정보인 분야 벡터와 입력 문서에서 작성된 문서 벡터 사이의 유사도 계산은 두 벡터가 이루는 각의 코사인 값을 이용하여 이하의 식과 같이 정의할 수 있다.

$$sim(D,\ Q) = \frac{D \cdot Q}{|D||Q|} \qquad \text{식 (9)}$$

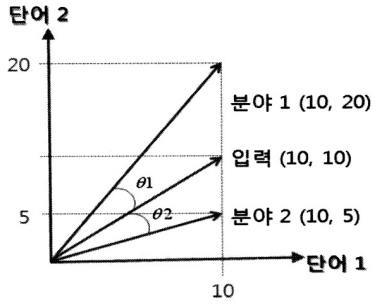
차원을 확장한 벡터에서 식 (9)를 이용하여 유사도를 계산한다. <표 6>을 확장하기 전의 분야 벡터에 대해 [분야1]에 대한 내용으로 작성된 문서가 입력되고 그에 해당하는 벡터는 <표 8>과 같이 유사도를 계산한다. 위의 (그림 13)에 입력 벡터와 분야 벡터가 벡터 공간 상에서 어떠한 관계를 갖는가 나타내었다. (그림 13)에서 $\theta 1$, $\theta 2$가 각각 입력 문서와 분야 1의 유사도, 입력 문서와 [분야 2]의 유사도가 된다.

〈표 8〉 입력 문서의 벡터

	단어1	단어2
입력 문서	10	10

〈표 9〉 확장한 입력 문서의 벡터

	단어1	단어2	단어1, 단어2
입력 문서	10	10	10



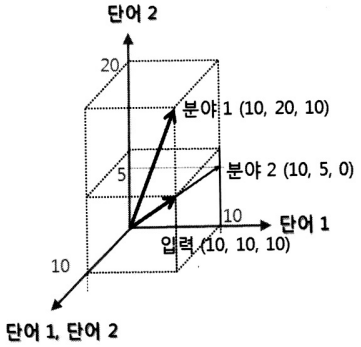
(그림 13) 입력 벡터와 분야 벡터와의 관계

입력 문서와 [분야 1], [분야 2]의 유사도를 계산하면 이하와 같은 결과가 얻어진다.

$sim([\text{분야}1],\ \text{입력}) = 0.9487$
$sim([\text{분야}2],\ \text{입력}) = 0.9487$

이 결과에서 알 수 있는 바와 같이 입력 문서는 [분야 1]. [분야 2]의 어느 쪽에도 적절한 분류라 할 수 없다. <표 8>에 표시한 입력 벡터에서 애매어와 해소어의 두 축을 증가시켜 차원 확장을 하면 <표 9>와 같다.

(그림 14)는 차원 확장을 한 입력 벡터와 분야 벡터의 벡터 공간 상에서의 관계를 표시하였다. 차원의 확장을 한 벡터를 이용하여 입력 문서와 [분야 1], [분야 2]의 유사도를 계산하면 다음과 같은 결과값이 얻어진다.



(그림 14) 입력 벡터와 분야 벡터의 관계

sim(입력, 분야1) = 0.9428
sim(입력, 분야2) = 0.7746

이 계산 결과에서 입력 문서는 [분야 2] 보다 [분야 1]에 해당하는 문서임을 알 수 있고, [분야 1]로 정확히 분류된다. 벡터의 차원을 확장하기 이전에는 어느 쪽으로 유사한지 판단할 수 없었으나, 벡터의 차원을 확장하면 정확하게 올바른 분야로 분류할 수 있음을 알 수 있었다.

5. 실험과 평가

5.1 실험 방법의 소개

본 논문에서는 제안한 방법의 유효성을 평가하기 위해 문서 분류 결과에 적용하는 실험을 하였다. 실험에 사용한 문서는 분류 문서와 입력 문서를 모두 언론 기사(네이버의 언론사별 뉴스)를 이용하였다. 사용한 데이터의 상세한 내용은 다음 절에 서술하였다. 실험에서 벡터 차원의 확장을 하지 않는 경우와 확장한 경우의 문서 분류에 관한 정확도를 비교하였다. 이하에 벡터 차원의 확장을 하지 않은 경우와 확장한 경우 각각의 문서 분류 순서를 제시하면 다음과 같다.

• 벡터의 차원 확장을 수행하지 않은 경우의 실험 순서

[순서-1] 각 분야의 문서에서 키워드를 추출한다. 분야 벡터를 작성한다.
[순서-2 입력 문서에서 키워드를 추출[5]하고, 입력 벡터를 작성한다.
[순서-3] 입력 벡터와 분야 벡터의 유사도를 계산한다.
[순서-4] 유사도가 높은 분야를 출력한다.

• 벡터의 차원 확장을 수행한 경우의 실험 순서

[순서-1] 각 분야의 문서에서 키워드를 추출하고, 분야 벡터를 작성한다.
[순서-2] 분류 정보에서 애매어의 추출한다.
[순서-3] 각 애매어에 대한 해소어를 결정한다.
[순서-4] 입력 문서에서 키워드 추출을 하고, 입력 벡터를 작성한다.
[순서-5] 각 벡터에서 차원을 확장한다.
[순서-6] 입력 문서에 애매어가 존재하는 경우에 벡터의 차원을 확장한다.
[순서-7] 입력 벡터와 분야 벡터의 유사도를 계산한다.
[순서-8] 유사도가 높은 분야를 출력한다.

애매어와 해소어를 결정할 때 임계값 α와 β를 사용하였으나, 최적의 값을 [스포츠] 분야에 적용하여 벡터 차원의 확장 전과 확장 후, 각각의 문서 분류 결과를 살펴보고 결정한다. 해소어의 정보를 상호정보량만으로 결정하는 방법과 공기의 세기까지도 고려한 방법 등 두 가지로 나누어 문서 분류의 결과에 어떠한 영향을 미치는지 조사한다. [스포츠] 분야를 통해 얻은 가장 좋은 결과를 임계값으로 사용하여 나머지 두 분야인 [경제], [정치] 분야에서 벡터 차원의 확장에 적용하고 그 정확도를 평가한다.

평가 방법으로는 벡터의 차원을 확장하지 않은 경우, 정답 분야가 첫 번째 후보로 검색되지 않아 잘못된 분류를 해버린 문서 중 벡터 차원을 확장하면 정답 분야의 검색 결과에 어느 정도의 향상을 가져오는가를 측정하였다. 종합적인 평가로서 정확률의 평균을 측정하였다. 정확률의 평균은 다음과 같이 계산한다. 정답 분야가 몇 번째에 검색되었는가를 결정하고 이때의 정확률을 계산한다. 입력 문서 전체에 대해 정확률을 계산하고 그 평균도 계산한다. 예를 들면, 정답 분야가 첫 번째에 검색되었을 때의 정확률은 100%이며, 두 번째에 검색된 경우에는 50%이며, 이 두 가지 정확률의 평균은 75%이다.

5.2 실험 데이터

본 논문에서 실시한 실험에서 이용한 문서는 인터넷 포털 사이트 네이버(Naver)/ 뉴스/ 언론사별 뉴스/ 각 91개의 사이트에서 추출한 [스포츠], [정치], [경제]에 대해 작성된 신문 기사에서 태그 정보를 확인하여 각각의 분야에 대해 세밀한 분야로 분류하여 태그 정보를 제외한 순수 텍스트 만을 사용하였다. 사용한 데이터의 자세한 사항은 앞의 <표 10>, <표 11>, <표 12>와 같다.

5.3 실험 결과

[스포츠], [경제], [정치] 분야를 사용하여 기존의 방법[6, 23, 28, 30을 이용한 분류 엔진의 구현한 시스템]에 의한 문서 분류의 결과를 <표 13> <표 14>, <표 15>에 각각 제시하였다.

<표 13>의 [스포츠] 분야에서 정답 분야가 검색된 순위를 살펴보면, 정확률 평균은 92.76%이고, 정답 분야가 1위로 검색되지 않은 문서 수는 106 개 이다. <표 14>의 [경제] 문야에서는 정확률 평균은 76.53%이고, 정답 분야가 1위로 검색되지 않은 문서 수는 149 개 이다. 그리고 <표 15>의 [정치] 분야는 정확률 평균은 85.01%이고, 정답 분야가 1위로 검색되지 않은 문서 수는 166 개 이다.

애매어와 해소어를 결정할 때 앞 장에서 설명한 바와 같이 임계값 α와 β를 각각 사용하였다. 이 임계값이 분류 결과에 어떤 영향을 주는가를 파악하고 최적인 임계값을 조사

〈표 10〉 [스포츠] 분야의 데이터 명세

분야명	분류 정보에 사용한 문서 수	입력 문서로 사용한 문서 수
복싱	27	28
검도	40	40
씨름	170	151
골프	40	40
마라톤	41	41
축구	163	100
농구	77	77
배구	95	95
럭비	79	79
야구	169	150
수영	22	22
트랙 경기	29	29
테니스	48	48
합계	**1,000**	**900**

〈표 11〉 [경제] 분야의 데이터 명세

컴퓨터 산업	40	42
주식/채권	50	37
금융-일반	40	37
경기-물가	40	33
경영	30	35
경제 이론	20	16
채용	50	30
국제 금융	30	20
세계 경제	70	80
한국 경제	50	50
무역	30	25
합계	**450**	**405**

〈표 12〉 [정치] 분야의 데이터 명세

외교	100	90
행정-내각	70	63
국회	50	45
정당	60	54
세제 혜택	60	54
선거	210	189
한국 경제	150	135
국방	100	90
합계	**800**	**720**

하기 위해 임계값 α와 β를 세 가지로 변화시키며 실험하였다.

<표 16>, <표 17>, <표 18>은 각각 해소어를 결정할 때의 임계값 β를 10%, 20%, 30%로 고정하고, 각 β의 값에 대해 애매어를 결정할 때, 임계값 α를 변화시켜 가며 계산한 문서 분류의 결과이다.

임계값 β를 고정하였을 때 어떤 결과에서도 $\alpha = 10\%$의 경우에 가장 좋은 결과가 얻어졌다. 이 때 임계값 $\alpha = 20\%, 30\%$ 일 때의 결과를 비교해 보면 변화가 보이지 않는다. 이 때 임계값 $\alpha = 10\%$ 일 때는 $\alpha = 20\%, 30\%$ 에 비해 검색 결과가 향상되고 있다.

임계값 $\beta = 10\%$ 에서 고정한 경우에 대해 생각해 보면 벡터 차원의 확장 대상이 되는 애매어의 수는 임계값

〈표 13〉 [스포츠] 분야에서 정답 분야가 검색된 순위

순위	문서 수	순위	문서 수
1위	794	8위	4
2위	55	9위	2
3위	24	10위	0
4위	12	11위	0
5위	4	12위	0
6위	4	13위	0
7위	1	-	-

〈표 14〉 [경제] 분야의 경우

1위	256	7위	6
2위	63	8위	0
3위	33	9위	1
4위	28	10위	0
5위	14	11위	0
6위	4	-	-

〈표 15〉 [정치] 분야의 경우

1위	554	5위	12
2위	65	6위	13
3위	41	7위	12
4위	22	8위	1

$\alpha = 10\%$ 일 때 739 개, 임계값 $\alpha = 20\%$ 일 때 969 개, 임계값 $\alpha = 30\%$ 일 때 981 개 이었다. 임계값 $\alpha = 20\%, 30\%$ 일 때 결과와 차이가 없는 이유는 벡터 차원의 확장 대상이 되는 애매어의 수에 차이가 없기 때문이라고 생각된다. 임계값 $\beta = 20\%, 30\%$ 의 경우도 같은 결과이었다.

임계값 $\alpha = 10\%$ 일 때 벡터 차원의 확장 대상이 되는 애매어는 39 개 이며, 임계값 $\alpha = 20\%, 30\%$ 일 때에 비해 그 개수가 적었다. 벡터 차원의 확장 대상이 되는 애매어의 수가 적으나, 분류 결과가 좋은 것은 다음과 같은 이유를 생각해 볼 수 있다. 임계값 $\alpha = 10\%$ 일 경우에 결정된 애매어를 "단어 1", 애매성을 갖는 분야를 [분야 1], [분야 2], 이에 반해 20%일 때 결정된 애매어를 "단어 2", 애매성을 갖는 분야를 [분야 3], [분야 4]로 하고, 각 분야에 출현하는 단어 수를 100 단어로 한정하였다. 상호정보량이 높은 단어의 순으로 본 경우, "단어 1"은 [분야 1]에서 첫 번째, [분야 2]에서 10번째의 경우 애매성이 약해졌으나, "단어 2"는 [분야 3]에서 첫 번째, [분야 4]에서 20번째 일 때 가장 애매성이 약해진다. 이 애매어의 세기가 다른 것은 같은 애매어로 처리하기 위한 것이라 생각된다. 결국 임계값 α는 선택된 애매어의 애매성의 세기를 결정하는 성질도 갖고 있다고 말할 수 있다.

다음에 임계값 α를 고정하여 임계값 β를 변화시킨 경우를 설명하면 다음과 같다. 임계값 $\alpha = 10\%$ 일 때는 변화가 보이지 않으나, 임계값 $\alpha = 20\%, 30\%$ 에서 임계값 β를 크게 하면 평균 정확률은 미세하게 향상을 보이나, 검색 결과가 좋은 문서의 수가 적어 결과가 나쁘게 나온 문서의 수도 감소하였다. 이 원인을 설명하면 다음과 같다. 임계값 $\alpha = 20\%$인 경우, 애매어 1,029 단어 중 임계값

〈표 16〉 임계값β= 10%, 임계값 α= 10, 20, 30 %의 결과

정답 분야로 검색되는 순위	α		
	10%	20%	30%
1위	813	798	798
2위	43	50	50
3위	19	25	25
4위	12	13	13
5위	3	4	4
6위	3	3	3
7위	2	1	1
8위	3	4	4
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.98%	92.97%	92.97%
결과가 향상된 문서 수	28	12	12
결과가 악화된 문서 수	3	7	7

〈표 17〉 임계값β= 20%, 임계값 α= 10, 20, 30 %의 결과

정답 분야로 검색되는 순위	α		
	10%	20%	30%
1위	813	798	798
2위	43	52	52
3위	19	23	23
4위	12	13	13
5위	3	3	3
6위	3	4	4
7위	2	1	1
8위	3	4	4
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.98%	93.00%	93.00%
결과가 향상된 문서 수	28	10	10
결과가 악화된 문서 수	3	4	4

〈표 18〉 임계값β= 30%, 임계값 α= 10, 20, 30 %의 결과

정답 분야로 검색되는 순위	α		
	10%	20%	30%
1위	813	798	798
2위	43	52	52
3위	19	23	23
4위	12	13	13
5위	3	3	3
6위	3	4	4
7위	2	1	1
8위	3	4	4
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.98%	93.00%	93.00%
결과가 향상된 문서 수	28	9	9
결과가 악화된 문서 수	3	3	3

$\beta = 10\%, 20\%, 30\%$ 의 경우 해소어가 보인 개수는 각각 969, 1018, 1021 단어 이었다. 결국 임계값 $\beta = 10\%$ 의 경우에 한정되지 않은 해소어가 조건을 낮게 조정하여 추출되는 것을 나타낸다. 임계값 $\beta = 10\%$ 의 경우에 추출된 해소어에 비해 임계값 $\beta = 20\%, 30\%$ 인 경우에 추출된 해소어는 상호정보량도 낮고, 애매어가 갖는 애매성을 해소하기에도 적당하지 않다.

공기의 강세를 고려한 해소어의 추출 방법에서도 애매어와 해소어를 결정할 때 각각 임계값 α 와 β를 사용하였다. 이 임계값이 분류 결과에 어떠한 영향을 주는가 또는 최적의 임계값을 조사하기 위해 임계값 α, β, γ 를 각각 변화시켜 실험하였다. 최적인 임계값 α , β를 구하기 위해 임계값 $\gamma = 100\%$ 로 고정하여 실험하였다.

<표 19>, <표 20>, <표 21>은 각각 애매어를 특정할 때의 임계값 α를 10, 20, 30%로 고정하고, 각 α값에 대해 해소어를 결정할 때의 임계값 β를 변화 시켜 문서 분류를 한 결과이다.

임계값 β를 고정해 보면, 어느 경우도 임계값 $\alpha = 10\%$ 일 때가 가장 좋은 결과가 얻어진다. 그 이유는 상호정보량 만으로 해소어를 결정한 경우와 같다고 생각된다. 또 임계값 α를 고정해 보면 어느 경우보다 임계값 $\beta = 30\%$ 로 한 경우에서 가장 좋은 결과를 얻었다. 왜냐하면 상호정보량이 높은 해소어는 애매어가 갖는 애매성을 해소하는 힘이 있다. 그러나 공기의 정도가 낮은 경우에는 입력 문서에 애매어와 해소어가 공기하여 출현하는 확률이 낮게 되어 그 힘이 발휘되지 않는다. 여기서 다소 애매성을 해소하는 힘이 약하여도 공기하는 경우가 강한 것을 해소어로 선택하여 입력 문서에 애매어와 해소어가 공기하여 출현하는 확률이 올라가고 결과적으로 문서 분류의 정밀도에 영향을 준다. 요약하면, 공기의 세기를 고려하여 해소어를 추출하는 경우에는 임계값 α를 낮게 하고, 임계값 β를 높게 하는 것이 바람직하다고 할 수 있다.

공기의 세기를 고려한 해소어 결정 시에 공기하는 세기의 정도를 표시하는 임계값 γ를 사용하였다. 이 임계값 γ가 문서 분류의 결과에 어떤 영향을 주고, 또한 최적값은 무엇인가를 확인하기 위해 가장 결과가 좋았던 임계값 $\alpha = 10\%$, 임계값 $\beta = 30\%$ 을 이용하여 임계값 γ를 90, 80, 70%로 변화 시켜 가며 실험하였다.

임계값 $\gamma = 100\%$ 일 때에 비해 각 임계값 γ 에서 검색 결과가 좋아지고 있다. 이러한 결과가 얻어지는 이유는 임계값 $\gamma = 100\%$ 으로 한 경우 해소어가 애매어와 완전히 공기하지 않으면 안 되므로 애매어에 대한 해소어가 분명하게 결정되어야 한다. 실제로 임계값 γ 가 100%, 90%, 80%, 70%로 변화 시킨 경우에 애매어 853 단어 중 해소어가 추출되는 것은 각각 699, 735, 800, 836 단어 이었다. 임계값 $\gamma = 80\%$ 에서 70%로 조정한 결과가 나빠지는 이유는 입력 문서에서 애매어와 해소어가 공기하여 출현하지 않을 확률이 높아졌기 때문이라 생각된다.

〈표 19〉 임계값 $\beta = 10\%$, 임계값 $\alpha = 10, 20, 30\%$의 결과

정답 분야로 검색되는 순위	α		
	10%	20%	30%
1위	804	795	794
2위	48	54	55
3위	23	24	24
4위	12	12	12
5위	2	4	4
6위	4	4	4
7위	1	1	1
8위	4	4	4
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.40%	92.81%	92.76%
결과가 향상된 문서 수	14	1	0
결과가 악화된 문서 수	0	0	0

〈표 20〉 임계값 $\beta = 20\%$, 임계값 $\alpha = 10, 20, 30\%$의 결과

정답 분야로 검색되는 순위	α		
	10%	20%	30%
1위	805	797	795
2위	47	52	54
3위	23	24	24
4위	12	12	12
5위	2	4	4
6위	4	4	4
7위	1	1	1
8위	4	4	4
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.46%	92.81%	92.81%
결과가 향상된 문서 수	15	3	0
결과가 악화된 문서 수	0	0	0

〈표 21〉 임계값 $\beta = 30\%$, 임계값 $\alpha = 10, 20, 30\%$의 결과

정답 분야로 검색되는 순위	α		
	10%	20%	30%
1위	805	797	796
2위	47	52	53
3위	23	24	24
4위	12	12	12
5위	2	4	4
6위	4	4	4
7위	1	1	1
8위	4	4	4
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.46%	92.81%	92.87%
결과가 향상된 문서 수	16	4	0
결과가 악화된 문서 수	0	0	0

〈표 22〉 임계값 $\alpha = 10\%$, $\beta = 30\%$, $\gamma = 90, 80, 70\%$의 결과

정답 분야로 검색되는 순위	$\gamma = 90\%$	$\gamma = 80\%$	$\gamma = 70\%$
1위	810	813	810
2위	45	40	44
3위	20	22	21
4위	12	11	11
5위	3	4	6
6위	4	3	1
7위	1	2	2
8위	3	3	3
9위	2	2	2
10위	0	0	0
11위	0	0	0
12위	0	0	0
13위	0	0	0
평균 정확률	93.80%	93.92%	93.78%
결과가 향상된 문서 수	27	28	28
결과가 악화된 문서 수	3	5	5

〈표 23〉 [정치] 분야의 결과

순위	문서 수	순위	문서 수
1위	559	5위	12
2위	60	6위	12
3위	46	7위	12
4위	18	8위	1

- 평균 정확률 : 85.43%
- 결과가 향상된 문서 수 : 16 문서
- 결과가 악화된 문서 수 : 4 문서

〈표 24〉 [경제] 분야의 결과

순위	문서 수	순위	문서 수
1위	248	7위	6
2위	61	8위	0
3위	43	9위	1
4위	27	10위	0
5위	15	11위	0
6위	4	-	-

- 평균 정확률 : 75.12%
- 결과가 향상된 문서 수 : 16 문서
- 결과가 악화된 문서 수 : 28 문서

〈표 25〉 [경제] 분야의 결과

순위	문서 수	순위	문서 수
1위	257	7위	6
2위	62	8위	0
3위	35	9위	1
4위	26	10위	0
5위	14	11위	0
6위	4	-	-

- 평균 정확률 : 76.70%
- 결과가 향상된 문서 수 : 3 문서
- 결과가 악화된 문서 수 : 0 문서

이상과 같이 애매어와 해소어를 결정할 때 사용하는 임계값 α, β, γ 의 최적값이 정해졌다. 가장 좋은 결과가 얻어져 상호정보량만으로 추출한 해소어를 이용한 경우의 임계값

$\alpha = 10\%$, 임계값 $\beta = 10\%$를 사용하여 [정치], [경제] 분야에서 분류 실험을 하였다. 각각의 결과를 표에 제시하였다.

기존의 방법으로 1위에 검색되지 않았던 문서 166 문서 중, 16개의 문서 검색 결과가 좋게 되어 약 9.6%의 정확률 향상이 확인되었다. 평균 정확률도 기존의 방법이 85.0%인 것에 비해 본 논문에서 제안하는 방법은 85.4%로 약간 향상되었다. 검색 결과가 나빠진 문서는 4개 정도 되었다.

기존의 방법에서 1위로 검색되지 않았던 문서 149개 중, 16개의 문서의 검색 결과가 좋게 나타난 반면, 검색 결과가 나빠진 문서가 28개 존재하였다. 평균 정확률도 기존의 방법이 76.5%인 것에 비해 본 논문의 방법은 75.1% 이었다. 그 원인은 다른 두 분야는 문서 정보가 되는 문서 수가 [스포츠] 1,000 문서, [정치] 800 문서인 것에 비해, 450개의 문서가 애매어나 해소어를 결정하기 위한 데이터가 적고, 각 단어가 갖는 정보량이 충분하지 않기 때문이라 생각된다. 공기의 세기를 고려하여 해소어를 추출하는 방법을 이용하고, 임계값 $\alpha = 10\%$, $\beta = 10\%$, $\gamma = 100\%$로 수정하여 실험하여 그 결과가 위의 <표 27>과 같다.

상호정보량만으로 결정한 해소어를 이용한 경우에 비해 검색 결과가 향상된 문서 수는 16개 문서에서 3개의 문서로 감소하였다. 결과가 나빠진 문서 수는 28개에서 0으로 그 수가 크게 감소하였다. 평균 정확률은 75.12%에서 76.70%로 향상하였다. 요약하면 분류 정보로 사용되는 문서 수가 많은 경우는 상호정보량만으로 해소어를 결정하는 것이 좋고, 반대로 문서 수가 적은 경우에는 공기 세기를 고려하여 해소어를 결정하는 것이 좋음을 알 수 있었다.

결론적으로 각 분야에서 기존 방법과 본 방법의 검색 결과가 변하지 않는 문서에 대해 조사하였다. 검색 결과가 변하지 않는 문서는 다음 두 가지의 경우로 나누어진다. 기존 방법과 본 논문에서 제안하는 방법에서 모두 정답 분야가 함께 첫 번째로 검색되는 경우와 첫 번째에 검색되지 않는 경우이다. 전자와 두 번째에 검색된 분야와의 유사도를 차이, 후자와 첫 번째로 검색되는 분야와의 유사도 차이의 평균을 구하였을 때 사용한 분류 결과는 각 분야에서 가장 좋은 결과가 얻어진 것을 사용하였다. 위의 <표 29>에 결과를 나타내었다. 이 표에서 각 분야에 대한 기존의 문서 분류 방법과 본 논문에서 제안하는 방법을 이용하여 정답 분야를 검색한 순위가 첫 번째 변화하지 않는 문서는 두 번째로 검색된 분야와의 유사도의 차이가 크게 나타나고 있다. 이것은 정답이 보다 정답답지 않다고 말할 수 있다. 기존의 방법, 본 논문의 방법에서 정답 분야의 검색된 순위가 두 번째 이후에서 변화되지 않은 문서는 첫 번째에 검색된 분야와의 유사도 차이가 적어진다. 순위는 바뀌지 않으나, 본 논문의 방법을 이용하면 정답 분야의 유사도가 다른 분야에 비해 큰 것을 알 수 있다.

〈표 26〉 [정치]의 결과가 향상된 문서의 순위에 대한 설명

기존 방법의 순위	본 논문에서 제안한 방법의 순위	문서 수
2위	1위	6
3위	2위	3
4위	3위	5
5위	4위	1
6위	5위	1

〈표 27〉 [경제]의 결과가 향상된 문서의 순위에 대한 설명

기존 방법의 순위	본 논문에서 제안한 방법의 순위	문서 수
2위	1위	5
3위	2위	3
4위	2위	1
4위	3위	3
5위	1위	1
5위	4위	3

〈표 28〉 [경제]에서 결과가 더 향상된 문서 조사

기존 방법의 순위	본 논문에서 제안한 방법의 순위	문서 수
2위	1위	1
4위	3위	2

6. 결 론

본 논문에서는 애매어와 해소어의 상호정보량을 벡터 공간 모델에 적용하여 문서 분류의 정밀도 향상에 관해 연구하였다. 벡터 공간 모델에 사용된 벡터는 같은 정도의 가중치를 갖는 축이 하나 더 존재하지만, 기존의 방법은 그 축에 아무런 처리가 이루어지지 않았기 때문에 벡터끼리의 비교를 할 때 문제가 발생한다.

같은 가중치를 갖는 축이 되는 단어를 애매어라 정의하고 단어와 분야 사이의 상호정보량을 계산하여 애매어를 결정하였다. 애매어를 갖는 애매성을 해소하는 단어를 해소어라 정의하고 애매어와 동일한 문서에서 출현하는 단어 중에서 상호정보량을 계산하여 결정하였다. 해소어에 의해 해소되는 애매어를 갖는 정보를 벡터 비교에 반영시키기 위해 애매어의 축을 "애매어"와 "해소어와 동시에 출현하는 애매어"라 하여 벡터의 차원을 확장하여 분류 정밀도를 향상시키는 방법을 제안하였다. 인터넷 포털사이트 네이버의 뉴스

〈표 29〉 검색 순위가 변화하지 않은 문서의 명세

	검색 순위가 첫 번째부터 변화하지 않는 문서		두 번째 이후부터 변화하지 않는 문서	
	기존 방법	본 논문의 방법	기존 방법	본 논문의 방법
[스포츠]	0.1935	0.2058	0.0596	0.0556
[정 치]	0.1625	0.1632	0.0957	0.0941
[경 제]	0.1381	0.1398	0.0903	0.0903

사이트에서 언론사별 뉴스를 수집하여 실험을 하고 그 유효성을 확인하였다.

향후의 연구로서는 애매성을 갖고 있는 분야의 정보 뿐 아니라 모든 분야의 정보를 고려한 해소어를 결정하는 방법에 대해 연구하고자 한다. 본 논문의 방법은 계산 속도를 고려하지 않았으므로 미래에는 처리의 고속화에 대해 연구하고자 한다.

참 고 문 헌

[1] 정경희, "의학 분야 웹 자료의 분류에 대한 개선 방안 연구", 정보관리학회지, 제21권, 제2호, pp.089-106, 2004.

[2] 윤성희, 백선욱, "단어 의미 정보를 활용하는 이용자 자연어 질의 유형의 효율적 분류", 정보관리학회지, 제21권, 제4호, pp.251-263, 2004.

[3] 이원휘, "K-Means 알고리즘을 이용한 대용량 문서 클러스트링에서 개선된 초기 중심 선정 방법의 제안", 전북대학교 대학원 컴퓨터공학과 박사학위 논문, pp.1-101, 2010.

[4] 안동언 외, 최신 정보검색론, 교보문고, pp.1-514, 2010.

[5] 이상곤 외, "개념 기반 복합 키워드 추출 방법", 한국컴퓨터교육학회 논문지, 제6권, 제2호, pp.23-31, 2003.

[6] 이상곤, "한글 문서 분류용으로 이용할 복합어로 구성된 분야 연상어의 추출법," 정보과학회 논문지: 소프트웨어 및 응용, 제32권, 제7호, pp.636-649, 2005.

[7] 노대욱 외, "정보 검색 기술을 이용한 비지도 학습 기반 문서 분류 시스템 개발," 정보과학회논문지 : 소프트웨어 및 응용, 제34권, 제2호, pp.123-130, 2007.

[8] 양재군, 배재학, 이종혁, "온톨러지 재사용을 위한 범주 재분류", 정보처리학회논문지(B), 제12권, 제1호, pp.69-80, 2005.

[9] 이원휘 외, "유해어 필터링과 SVM을 이용한 유해 문서 분류 시스템," 정보처리학회논문지(B), 제16권, 제1호, pp.85-92, 2009.

[10] 박흠, "확장된 Relief-F 알고리즘을 이용한 소규모 크기 문서의 자동 분류", 정보처리학회논문지(B), 제16권, 제3호, pp.233-238, 2009.

[11] 김판구 외, "상호 정보에 기반한 한국어 텍스트의 복합어 자동색인," 한국정보과학회 논문지, 제21권, 제7호, pp.1333-1340, 1994.

[12] 김명철 외, "시소러스와 상호 정보를 이용한 정보검색 모델", 한국정보과학회 학술발표 논문집, 제21권, 제1호, pp.837-840, 1994.

[13] 전미선 외, "상호 정보를 이용한 어의 모호성 해소에 관한 연구", 제 6회 한글 및 한국어 정보처리 학술발표 논문집, pp.369-373, 1994.

[14] 강현규 외, "자연언어 정보검색에서 상호정보를 이용한 2단계 문서 순위 결정 방법", 한국정보과학회 논문지, 제23권, 제8호, pp.852-861, 1996.

[15] 강현수 외, "정보 검색에서 상호 정보를 이용한 용어 확장 및 한정 연구", 한국정보과학회 호남·제주지부 학술발표 논문집, 제10권, 제1호, pp.128-134, 1998.

[16] 이찬도 외, "고품질 바이그램을 이용한 문서 범주화 성능 향상," 정보처리학회 논문지 B, 제9-B권, 제4호, pp.415-420, 2002.

[17] 최준영 외, "효율적인 바이어그램을 이용한 자동 문서 범주화," 제 19회 한국정보처리학회 춘계 학술대회 논문집, 제10권, 제1호, pp.261-264, 2003.

[18] 박은석, 박현진, 이상곤, "동의어와 유의어 개념에 기반 한 키워드 추출기의 설계 및 구현", 컴퓨터종합학술대회 2007 논문집, 제34권, 제1(C)호, pp.163-166, 2007.

[19] 장정효, 손주성, 이상곤, 안 동 언, "연상 지식을 이용한 문서 분류 엔진의 구현", 제25회 정보처리학회 춘계 학술발표대회 논문집, 제13권, 제1호, pp.625-628, 2006.

[20] 장정효, 손주성, 김도연, 이상곤, 이원휘, 안동언, "검색과 분류가 동시에 가능한 JULSE 시스템의 설계 및 구현", 제24회 정보처리학회 추계 학술발표대회 논문집, 제12권, 제2호, pp.673-676, 2005.

[21] 김혜경, 이상곤, "화제인식에 의한 단락별 계산방법의 설계", 컴퓨터종합학술대회 2005 논문집, 제32권, 제1(B)호, pp.499-501, 2005.

[22] 임수정, 이원휘, 이상곤, "화제출현, 계속, 전환 처리를 이용한 한국어 문서의 단락분할", 제23회 정보처리학회 춘계 학술발표대회 논문집, 제12권, 제1호, pp.737-740, 2005.

[23] 이상곤, "분야연상어를 이용한 화제분야의 계산방법과 단락검색", 정보처리학회논문지(B), 제12권, 제1호, pp.57-68, 2005.

[24] 이원휘, 김도연, 이상곤, "그래픽컬한 분야인식기의 설계 및 구현", 정보과학회 가을 학술발표 논문집, 제31권, 제2호, pp.769-771, 2004.

[25] 이원휘, 최현, 이상곤, "분야연상어 추출방법의 설계와 구현", 정보처리학회 2004년도 춘계 학술발표 논문집, 제11권, 제1호, pp.651-654, 2004.

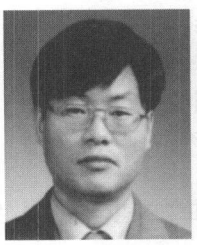
[26] 최현, 황남선, 이상곤, "문서분류용 목적으로 이용할 효율적인 연상정보의 추출방법", 2004년 봄 정보과학회 학술발표 논문집(B), 제31권, 제1호, pp.892-894, 2004.

[27] 김양선, 이상곤, "단어개념에 기반 한 한국어 복합키워드의 추출", 제20회 한국정보처리학회 추계 학술발표 논문집, 제10권, 제2호, pp.477-480, 2003.

[28] 이상곤, 이완권, "분야연상어의 수집과 추출 알고리즘", 정보처리학회 논문지(B), 제10권, 제3호, pp.347-358, 2003.

[29] 홍성옥, 이상곤, "연상정보를 이용한 단락분할 방법", 2003년도 정보처리학회 춘계 학술발표 논문집(상), 제10권, 제1호, pp.497-500, 2003.

[30] 이상곤, "분야연상어를 이용한 화제의 계속성과 전환성을 추적하는 단락분할방법", 정보처리학회 논문지(B), 제10권, 제1호, pp.57-66, 2003.



이 상 곤

e-mail : samuel@jj.ac.kr

1994년 전주대학교 영어영문학과(이학사)

1996년 전북대학교 컴퓨터과학과(이학사)

1998년 전북대학교 전산통계학과(이학 석사)

2001년 日本 도쿠시마대학교 지능정보공학과(공학 박사)

2002년 원광대학교 음성 정보 기술 산업 지원 센터 연구원

2002년~현 재 전주대학교 컴퓨터공학과 부교수

관심분야 : 한국어 정보 처리, 한글 공학, 정보검색, 문서 분류 및 요약, 키워드 추출, 컴파일러, 인공지능에 대한 연구