한영 혼용문에서 괄호 안 대역어구의 자동 인식

이 재 성† · 서 영 훈††

요 약

한영 혼용문에서 번역된 전문용어 등을 사용할 때, 이해를 돕기 위해 그 뒤의 괄호 안에 원어 풀이를 함께 쓰는 경우가 많다. 본 논문에서는 괄호가 사용된 구가 대역어구 관계인지를 판단하고, 어느 범위까지 대역어구인지를 기본사전을 이용하여 확률적으로 계산하고 인식하는 방법을 제시한다. 특히, 사전에 표제어로서 혹은 대역어로서 존재하지 않는 단어들을 처리하기 위해 음운유사도 일치, 대역어 부분일치의 방법과 복합어 처리를 위해 부분일치 방법을 새로 제안하였다. 각 방법들을 단계별로 실험하여 0.4F값(α를 0.4로 설정한 F값)으로 측정한 결과, 기본 실험 방법인 사전 대역어 완전일치방법의 경우 23.8%인데 비해, 대역어 부분일치와 음운유사도 일치를 혼합한 방법이 75.9%, 복합어 처리를 추가한 방법이 77.3%의 값을 보여 성능이 최고 3.25배 향상되었다.

Automatic Recognition of Translation Phrases Enclosed with Parenthesis in Korean-English Mixed Documents

Jae Sung Lee† · Young Hoon Seo††

ABSTRACT

In Korean-English mixed documents, translated technical words are usually used with the attached full words or original words enclosed with parenthesis. In this paper, a collective method is presented to recognize and extract the translation phrases with using a base translation dictionary. In order to process the unregistered title words and translation words in the dictionary, a phonetic similarity matching method, a translation partial matching method, and a compound word matching method are newly proposed. The experiment result of each method was measured in F-measure(the alpha is set to 0.4) ; exact matching of dictionary terms as a baseline method showed 23.8%, the hybrid method of translation partial matching and phonetic similarity matching 75.9%, and the compound word matching method including the hybrid method 77.3%, which is 3.25 times better than the baseline method.

키워드 : 자동추출(automatic extraction), 정렬(alignment), 대역어 부분일치(translation partial matching), 복합어 일치(compound word matching)

1. 서 론

영어와 한글을 많이 혼용하여 사용하는 과학 기술 등의 논문에서는 번역된 전문용어 뒤에 대개 괄호를 사용하여 영어를 표기함으로써 그 뜻을 명확하게 설명한다. 이런 정보는 논문 내에서 하나의 단위로 중요한 개념을 전달하는 경우가 많으며, 한 단위로 파싱하거나 검색되는 경우, 좀더 효과적일 수 있다. 따라서 이를 추출하면, 그 문서의 분석에 즉시 이용하거나, 장기적으로 대역어 사전을 구축하여 교차언어(cross language) 정보검색이나 기계번역(machine translation)에 이용할 수 있다. 특히, 다어구로 된 전문용어의 경우, 대역어 사전 구축이 어려우므로, 자동인식을 통한 추출 방법이 필요하다[1].

† 종신회원 : 충북대학교 컴퓨터교육과/컴퓨터 정보통신연구소 교수
†† 종신회원 : 충북대학교 컴퓨터공학과/컴퓨터 정보통신연구소 교수
논문접수 : 2002년 5월 3일, 심사완료 : 2002년 7월 21일

괄호 안의 대역어를 인식하여 추출하기 위해서는 첫 번째로 괄호 안의 내용과 괄호 앞의 대응어가 대역어 관계인지를 파악해야하고, 두 번째로 괄호 앞의 어절이 어느 범위까지 대응되는지를 판단할 수 있어야한다. 본 논문에서는 이 두 단계 구현을 위해 이중언어 정렬방법을 사용한다[2-4]. 즉, 괄호 앞의 어절과 괄호 안의 어절을 최대의 확률로 정렬하고, 그 확률값이 일정한값 이상일 경우, 대응되는 대역어구로 인식하며, 이때 서로 정렬이 이루어진 어절들까지를 대응 대역어구로 추출한다.

이중언어 정렬(bilingual alignment)은 주로 대량의 코퍼스(corpus)로부터 대역어 사전을 구축하는 방법으로 사용되었다[2-4]. 이러한 방법으로는 사전을 전혀 사용하지 않고 순수한 통계적 방법으로 처음부터 구축하는 방법[2-4]과 초기에 사전의 정보를 이용하는 방법[5], 동족어의 철자 유사성을 이용하는 방법[6] 등, 다양한 시도가 이루어지고 있다. 괄

호 대역어 추출을 위해서는 주로 짧은 문맥내에서 괄호를 인식하고 괄호 안과 괄호 앞에 사용된 어절들을 분석해야 하기 때문에 사전이나 언어의 유사성 등과 같은 외부 정보가 없으면 추출이 어렵다. 본 논문에서는 이를 위해 기본 대역어사전을 사용하였다. 사전을 사용할 경우, 주로 문제되는 것은 미등록어이며, 이를 보완하기 위해 음운유사도를 계산하여 음차표기(transliteration)된 외래어 구절을 추출할 수 있도록 했다. 또한 사전에 등록된 단어이더라고, 사전에 있는 대역어와 실제 사용된 대역어가 일치하지 않을 경우, 정렬이 되지 않으므로 이 문제를 해결하기 위해 대역어의 부분일치를 확률적으로 계산하여 새로운 대역어를 추출할 수 있도록 했다. 또 복합어에 대한 부분일치를 할 수 있도록 하여, 사전의 단어 혹은 음차표기가 서로 복합어 형태로 사용되더라도 추출할 수 있는 방법을 제시하였다.

논문의 순서로서 우선 2장은 관련연구로서 통계적 정렬에 대해 소개하고, 3장에서는 괄호 안에 사용된 대역어들의 용례를 분석한다. 또, 4장에서는 구체적인 대역어 추출 알고리즘을 설명하고, 5장에서 실험 내용 및 결과, 6장에서 실험 결과에 대한 분석 및 토의를 하고 7장에서 결론을 맺는다.

2. 관련 연구

이중언어 정렬은 많은 양의 원문과 번역문이 존재할 경우, 이들로부터 실제 사용되는 번역어를 추출하고, 이를 통계적으로 분석하여 자동으로 기계번역을 하기 위한 시도이다. 이러한 연구는 Brown[2, 3]에 의해 처음 시도되었고, 캐나다 국회에서 영어와 불어로 기록된 많은 양(영어 약 350만 문장, 불어 약 370만 문장)의 이중언어 코퍼스(Hansard)를 사용하여 시도되었다. 즉, 번역문장내에서 대역단어는 서로 대응되는 대역 문장쌍에서 자주 공기(co-occurrence)한다는 사실과 문장내 위치와 연관성이 있음을 이용하여 대역단어의 정렬을 하였다. 이는 충분히 많은 양의 이중언어 문서가 있어야 의미있는 통계적 분석이 가능하다.

Brown[2, 3]의 연구는 주로 초기부터 순수하게 통계적인 정보에 의존하여 확률값을 가진 대역어 사전을 구축하고 이 사전을 반복 사용하여 점차 더 정확한 확률값을 계산하도록 하였다. 하지만, 코퍼스의 내용에 오류가 포함되거나 구조가 매우 다른 언어사이에서의 정렬은 초기의 확률값 계산이 부정확하여 정확한 대역어 사전 구축에 어려움이 있다. 이러한 문제점들을 해결하기 위한 시도가 Church[6]와 Wu[5, 7]등에 의해 이루어졌다.

Church[6]은 영어와 불어가 동족어로서 같은 철자들을 많이 공유하는 특징을 이용하였다. 예를 들어 영어 단어 "government"는 같은 의미의 불어 단어 "gouvernement"와 유사한 철자가 많이 있다. 이러한 특징을 이용하여 단어를 4-gram(4글자씩 분리된 단위)으로 분리하고 공통적인 4-gram이 많은 단어들은 대역어로 판별함으로써 오류가 많은 문서에 대해서도 견고하게 작동하였다.

Wu[5, 7]는 영어-중국어 쌍에 대해 정렬을 시도하였는데, 영어가 띄어쓰기를 하는데 반해, 중국어는 띄어쓰기를 하지 않으므로 정렬의 효율을 위해 중국어 띄어쓰기가 전처리로 필요했다. 띄어쓰기는 사전에 등록된 단어 단위로 분리하였다. 하지만, 미등록된 단어가 많았으므로 이를 해결하기 위해 텍스트로부터 단어들의 공기 정보를 추출한 후, 이 단어들을 수작업으로 선별하여 새로운 표제어를 추가하였다. 이렇게 미등록 문제를 어느 정도 해결함으로써, 영어-중국어의 정렬 성능을 향상시켰다.

신중호[8]는 한국어와 영어의 정렬을 시도했으며, 언어의 구조적 차이를 극복하고, 이중언어 코퍼스 양의부족을 해결하기 위해, 형태소해석기 및 태거를 전처리기로 사용하였다. 즉, 정렬 이전에 단어들을 각 품사별로 구분하여 이 정보를 이용하여 정렬의 효과를 높였다.

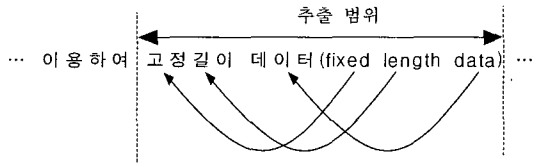
본 논문에서는 한국어 문서내에서 사용된 괄호를 중심으로 대역어를 인식하여 추출하는 것이다. 이를 위해, 괄호 안의 단어들이 괄호 앞의 단어들과 정렬이 되는지를 일정한 범위내에서만 판정하면 된다. 따라서, 기존의 이중언어 정렬과 같이 많은 양의 코퍼스 정보를 전혀 이용할 수 없다. 대신, 적은 양의 단어들을 판별할 수 있도록 음운유사도 정보나 사전 정보를 이용할 수 있다. 그러나 사전 정보를 이용할 경우, 미등록어 문제가 발생할 수 있다. 특히, 대역사전일 경우, 미등록어는 표제어에 국한되지 않고, 사전 내용 중의 미등록 대역어도 있을 수 있다. 또한, 단순 단어들은 사전에 있지만, 복합어 형태로 사용될 경우, 미등록어로 처리되는 경우도 있다. 이러한 문제들은 대역어 추출의 성능을 매우 감소시키는데, 본 논문에서는 이러한 문제들을 어느 정도 해결하여 대역어 추출 성능을 획기적으로 향상시킨 방법을 제시한다.

3. 괄호 대역어 사용예 분석

괄호는 문서 내에서 여러 가지 형태로 사용되고 있다. 즉, 괄호 앞의 내용을 좀더 서술하기 위해서 사용되는 경우, 약어를 풀어 쓴 경우, 동의어를 나타내기 위해서 사용되는 경우(이 경우는 같은 언어로 동의어를 사용하는 경우와 대역어로 같은 뜻을 사용하는 경우가 있다.), 하나의 분리 단위로 나타내기 위해서 단순히 사용되는 경우, 수식의 일부로 사용되는 경우 등이 있다. 본 논문에서는 그 중에서도 한글 어구 뒤에 괄호와 함께 영어 단어구를 사용한 형태에 대해서만 처리한다. 괄호 안에 사용된 영어 단어구에는 경우에따라 약어 등이 함께 표시되는데 이는 전처리를

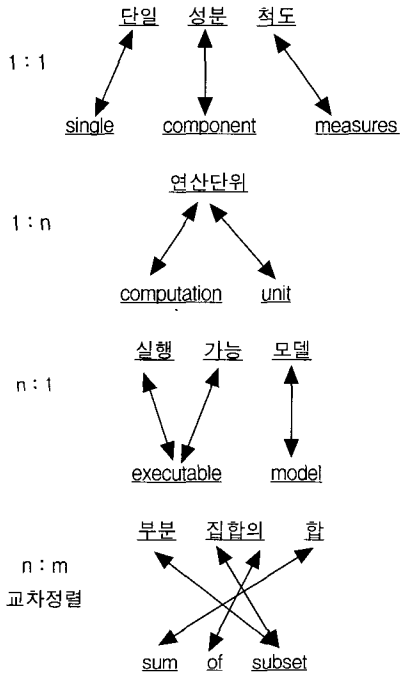
통해 미리 분리해 낼 수 있다. 따라서, 본 논문에서는 한글 어구 뒤에 영어 단어구가 사용된 형태만을 처리하기 위한 방법을 집중적으로 설명한다.

괄호 어구가 대역어인지를 판별하기 위해서는 괄호 안의 각 단어들이 괄호 앞의 단어들과 대역어 관계인지를 파악해야 한다. 예를 들어 "… 이용하여 고정길이 데이타(fixed length data)"라는 어구가 있을 경우, (그림 1)과 같이 대응 관계가 있음을 확률적으로 파악하고, 그에 따른 대응 범위도 결정하여 대역어구를 인식한다. 즉, "fixed"가 "고정", "length"가 "길이", "data"가 "데이터"에 정렬되는 것을 알아내어 "fixed length data"가 앞의 "… 이용하여"를 제거하고 "고정길이 데이터"만이 대응되는 어구임을 판단해야 한다.



(그림 1) 대역어구 추출 범위 예

대역어의 대응 유형을 살펴보면, 한글 어절과 영어 단어가 1 : 1로 대응되는 경우, 한글 어절 1개가 영어 단어 n개에 대응하는 1 : n의 경우, 한글 어절 n개가 영어 단어 1개와 대응하는 n : 1의 경우가 있다. 1 : n의 경우, 한글에서 명사사이의 띄어쓰기가 비교적 자유롭기 때문에 한글 명사를 붙여써서 발생된 형태이며, n : 1은 드물지만, 한 영어 단어를 풀어써서, 한글이 여러 어절로 대응되는 경우이다.



(그림 2) 정렬의 다양한 예

또, 대응 순서가 뒤바뀌어 나타나는 경우도 있다. 즉, (그림 2)에서 보인 예처럼, "부분 집합의 합"과 "sum of subset"의 경우, 한글의 첫 어절 및 둘째 어절중 일부인 "부분 집합"이 영어의 3번째 단어인 "subset"과 대응되고, 한글의 3번째 어절인 "합"이 영어의 첫 번째 단어인 "sum"과 교차되어 대응되는 것을 보여 준다. 따라서, 대역어 확인을 위해 일단 사전정보나 음차표기 정보등을 통해 대역관계를 정보를 알아내고, 이를 이용하여 각 단어들의 대역관계와 복합어들의 관계를 확인해야 하며, 순서대로 된 단어들과 순서가 뒤바뀐 단어들에 대해서도 대역 관계를 확인해야 한다.

4. 대역어 추출 알고리즘

4.1 사전번역확률의 계산

사전번역확률은 문서에 있는 원어(이하 문서원어)와 대역어(이하 문서대역어)가 사전에 있는 등록되어 있는 원어(이하 사전원어)와 대역어(이하 사전대역어)와 일치하는지를 검색하여 계산된다. 일반적으로 사전검색은 단어들의 완전일치만을 고려하여 처리하지만, 본 논문에서는 부분일치도 고려하여 대역어 선택의 가능성을 높였다.

완전일치의 경우는 문서원어와 문서대역어가 사전에 존재할 경우, 확률을 1로 보고, 그 이외의 경우는 확률 0으로 계산한다. 완전일치는 유사한 단어나 어휘상으로 약간의 차이가 있는 단어들까지도 전혀 다른 단어로 보아 모두 확률 0으로 처리하므로, 다양하게 사용된 어휘들의 대역관계를 처리하기 힘들다. 이러한 문제점을 해결하기 위해 부분일치를 사용할 수 있으며, 본 논문에서는 문서원어를 원형복원하여 다시 완전일치를 시도하는 방법과 음절정보를 이용하여 문서대역어와 사전대역어들의 부분일치 정도를 계산하는 방법을 사용한다.

대역어의 부분일치는 대역어 사전에서 대개 완벽하지 못하므로 이를 보완하기 위해, 이미 있는 사전대역어의 특성을 이용하기 위한 것이다. 예를 들어 "query"의 경우, "질문", "의문"은 사전 대역어로 존재하지만, "질의"는 없었다. 그러나, "질의"라는 단어는 질문과 의문이라는 단어와 유사한 뜻이고 어휘적으로도 두 단어들의 첫 글자를 합치면 "질의"와 일치한다. 어원을 따져볼 경우, 한자가 같기 때문에 같은 뜻으로 볼 수 있다. 이러한 유사성을 계산하기 위해 본 논문에서는 음절단위의 일치정도를 계산하였다. 즉, 식 (1)과 같이 각 단어에서 일치하는 음절(syllable) 수를 각 단어에 나타난 유일한 음절수(중복되어 나타난 음절을 1개로 계산)로 나누어 대역어의 유사도를 계산하였다. 대역어 부분 일치 확률은 가능한 대역어중 가장 확률이 높은 것을 구해야 하며 이는 식 (2)와 같다. 예를 들어, TSIM("query", "질의")를 본 논문에서 사용한 대역어사전을 기반으로

계산하면, MAX(WSIM("질문", "질의"), WSIM("의문", "질의"))로 되고, 이를 소수점 첫째자리까지 계산하면 MAX(0.3, 0.3)이 되어 0.3의 값을 갖는다.

$$WSIM(w_i, w_j) = \frac{w_i \text{와 } w_j \text{의 공통음절수}}{w_i \text{와 } w_j \text{의 음절수(중복제외)}} \quad (1)$$

$$TSIM(E, K) = MAX_i WSIM(T_i, K) \quad (2)$$
(T_i 는 E 의 한국어 사전대역어중의 하나)

4.2 음차번역확률의 계산

음차번역확률은 기본적으로 사전에 등록되지 않은 단어를 음차된 단어로 보고, 이 확률을 계산하는 것이다. 음차표기 확률은 두 단어 사이의 음운유사도 계산을 이용할 수 있으며, 영어단어와 외래어로 표기된 한국어 단어쌍으로부터 자동으로 학습된 각 발음단위의 정렬확률을 이용하여 계산될 수 있다[9, 10]. 즉, 영어와 한국어를 각 글자 단위 혹은 발음단위(음차표기시 한 단위로 표기되는 글자들)로 분리한 후, 이들 사이의 정렬 가능성을 확률로 계산한 후, 이 값들을 전체 단어에 대해 계산하여 확률로 나타낼 수 있다. 이를 식으로 나타낸 것이 식 (3)이며, 여기에서 E와 K는 영어단어와 한국어단어를 각각 나타내며, EU와 KU는 영어발음단위와 한국어발음단위를 나타낸다. 이 식에서는 각 발음단위의 대응확률을 모두 곱한 후에 이를 다시 발음단위 개수로 정규화하여 확률을 계산하였다.(일반적으로 두 단어사이의 유사도는 순서에 관계없이 계산되나, 식 (3)에서는 영어가 한글로 표기된 것을 기준으로 하였기 때문에 조건확률에서 영어와 한글의 순서가 고려되었다.) 예를 들어, PSIM(data, 데이터)의 값을 계산하려면 우선 P(data | 데이타)로 바꾼 후, 이를 각 발음단위 대응확률 곱인 P(dl ㄷ)×P(a | ㅔ이)×P(t | ㅌ)×P(a | ㅏ)로 계산하고, 이를 발음단위 갯수인 4로 정규화하여 4제곱근의 수를 구하면 된다. 이때, 발음단위로 분리되는 방법은 여러 가지가 있으나 그 중에서 정렬확률 즉, PSIM(E, K)가 최대의 값을 가질 수 있도록 분리한다.

$$PSIM(E, K) = P(E \mid K) = MAX \sqrt[n]{(\prod P(KU_i \mid EU_i))} \quad (3)$$

4.3 한국어 복합어의 부분일치

한국어의 경우, 대개 띄어쓰기가 고정적이지 않아 복합어를 여러 가지로 띄어 쓰는데 반해, 영어의 경우, 대부분 명확하게 분리하여 쓴다. 이러한 용례를 이용하여, 영어단어를 기준으로 한국어 복합어에 대해 부분일치를 시도하고, 그 중 일치 확률이 최대값을 갖는 부분 문자열을 대역된 문자열로 판단한다. 식 (4)와 식 (5)는 각각 사전번역 확률과 음차번역 확률을 복합어 부분일치 방법으로 다시 표기한 것이다. 여기에서 $K_{j,k}$는 한국어 K에서 j번째 음절부터 k번째 음절까지의 부분문자열을 나타내며, K의 음절수가 n일 경우, $1 \le j \le n$, $j \le k \le n$의 관계식이 성립한다.

$$TCSIM(E, K) = MAX_{j,k}\ TSIM(E, K_{j,k}) = MAX_{i,j,k}\ WSIM(T_i, K_{j,k}) \quad (4)$$

$$PCSIM(E, K) = MAX_{j,k} PSIM(E, K_{j,k}) \quad (5)$$

<표 1>은 영어단어 "working"이 한국어 "작업메모리"와 부분일치 확률을 사전번역을 통해 계산하는 과정을 보여준다. 단, 사전에는 "working" 표제어에 "작업"이라는 대역어가 포함되어 있다. 부분일치는 우선 TSIM("working", "작")에 대해 수행되고 그 결과는 표의 2행 1열에 0.5로 표시된다. 다음으로 TSIM("working", "작업")에 대해 수행되고 결과가 2행 2열에 1.0으로 표시된다. 계속하여 TSIM("working", "작업메")를 수행하여 0.66667값을 넣고 같은 방법으로 2행을 모두 계산한다. 같은 방법으로 3행은 "working"과 "업메모리"의 부분문자열에 대해 각각의 유사도 계산하고, 4행, 5행, 6행도 "메모리", "모리", "리"에 대해 유사도 계산한다. 최종적으로 가장 높은 유사도를 가진 TSIM("working", "작업")에서 사용된 부분문자열 "작업"을 대역어로 선택한다.

또 <표 2>는 같은 방법으로 "memory"와 "작업메모리"에 대해 음차번역 확률을 계산한 결과이다. 이 경우, 4행 5열이 0.415739로 가장 높은 확률을 보여 "작업메모리" 중에서 "메모리"가 "memory"와 부분일치함을 보여준다.

〈표 1〉 "working"과 "작업메모리"의 부분사전번역확률

작	업	메	모	리
0.500000	**1.000000**	0.666667	0.500000	0.400000
	0.500000	0.333333	0.250000	0.200000
		0.000000	0.000000	0.000000
			0.000000	0.000000
				0.000000

〈표 2〉 "memory"와 "작업메모리"의 부분음차번역확률

작	업	메	모	리
0.000043	0.000318	0.000278	0.000133	0.000083
	0.000041	0.000354	0.004520	0.000980
		0.001151	0.050972	**0.415739**
			0.001099	0.111149
				0.000061

4.4 추출 알고리즘

괄호 안에 사용된 용어가 대역어 관계인지를 파악하기 위해서는 <표 3>과 같은 알고리즘을 사용한다. 즉, 우선 1단계로 괄호가 사용된 문맥을 추출한다. 이는 간단한 오토마타를 작성하여 열린 괄호가 시작되는 부분과 닫힌 괄호가 나오는 부분을 찾아내고, 괄호 안의 어절이 대응될 수 있는 정도의 크기로 괄호 앞 어구를 추출한다. 2단계로는 이 어구내에서 약어가 있는지를 판별하고 추출한다. 3단계에서 약어를 제외한 후, 괄호 안의 어구가 남아 있을 경우, 괄호 안의 어구와 괄호 앞의 어구를 다시 배열하여 대역어 판별을 할 수 있도록 잠정어구를 구성한다. 4단계로 각 어

절에 대해 괄호 안의 어절과 괄호 앞의 어절이 대역어 관계인지를 계산한다. 즉, 대역어 사전을 찾아서 대역 관계를 판단하는 사전번역확률과 음운유사도를 이용한 음차번역확률을 계산한다. 이때 순서에 관계없이 교차정렬 등을 찾아내기 위해, 각 어절에 대해 가능한 모든 대응 어절들에 대해 계산하며 이중 가장 큰 사전번역확률 혹은 음차번역확률을 계산하고, 또 다시 이 두 가지 확률 중에 큰 값을 어절번역확률로 계산한다. 5단계에서는 이미 계산된 어절번역확률을 모두 곱하여 어구번역확률을 계산한다. 6단계에서는 이 어구번역확률을 근거로 그 값이 일정한 값 이상이면, 대역어관계로 파악하고, 그에 대응되는 대역어구를 추출한다. 이 과정에서 괄호 앞 어구와 괄호 안의 내용이 대역 관계가 아닌 설명관계이거나 수식의 기호 등인 경우, 한글과 영어로 대응되는 대역확률은 거의 0이거나 0에 가까운 값을 갖게 되어 자동으로 제거된다.

$$SIM(E,K) = MAX(\ TSIM(E,K),PSIM(E,K)) \qquad (6)$$

〈표 3〉 괄호 대역어 추출 알고리즘

1. 괄호가 사용된 문맥 추출
2. 문맥 내에서 사용된 약어가 있는지 검사 및 추출
3. 약어를 제외한 괄호 안과 앞의 잠정어구 추출
4. 각 어절에 대해 원어절과 대역어절 확률계산
   4-1. 사전번역확률 계산
   4-2. 음차번역확률 계산
   4-3. 어절번역확률 선택
5. 어구번역확률 계산 : 모든 어절번역확률의 곱
6. 일정값 이상의 어구번역확률로 정렬된 어구 추출

5. 실 험

실험에 사용된 영한 대역어사전은 수작업으로 만든 사전으로 총 67,600여개의 영어 표제어와 표제어당 평균 3.3개의 대역어를 가지고 있다. 또, 음운유사도 계산을 위해서 1,500여 영어-외래어 단어쌍에서 추출된 발음단위 음차 확률을 사용하였다[9].

실험대상은 정보검색 테스트 집합인 KTSET 1.0을 이용하였다[11]. KTSET 1.0은 1,000개의 과학논문 요약으로 구성되어 있다. 이중 한국어 요약부분에서 괄호를 사용한 용례를 추출하여 그 정확도를 측정하였다. 실험의 편의상 1에서 150번까지의 요약에 대해서만 평가를 하였다.

평가는 대역어구인지를 정확히 판단하여 정확한 대응 어구를 분리한 것만을 맞는 것으로 했으며, 앞 절에서 제안한 각 방법에 대해 그 효과를 측정하였다. 성능 측정의 기준은 정보검색에서 흔히 사용되는 재현률 식 (7)과 정확률 식 (8) 및 F값 식 (9)을 사용하였다. F값은 재현률과 정확률에 대한 가중치를 α변수에 부여하여 계산할 수 있는데, 재현률과 정확률을 같은 비율로 고려할 경우, α값은 0.5이다. 이를 "0.5F값"이라고 편의상 사용한다. 본 실험에서 대역어 추출 후 약간의 오류는 수작업으로 수정할 수도 있음을 고려하여 재현률에 대해 가중치를 좀더 부여한 0.4F값을 계산하였다.

$$\text{재현률}(R\ :\ Recall) = \frac{EM}{EM+NE} \qquad (7)$$

$$\text{정확률}(P\ :\ Precision) = \frac{EM}{EM+FE} \qquad (8)$$

$$F\text{값} = \frac{1}{\alpha \times \frac{1}{P} + (1-\alpha) \times \frac{1}{R}} \qquad (9)$$

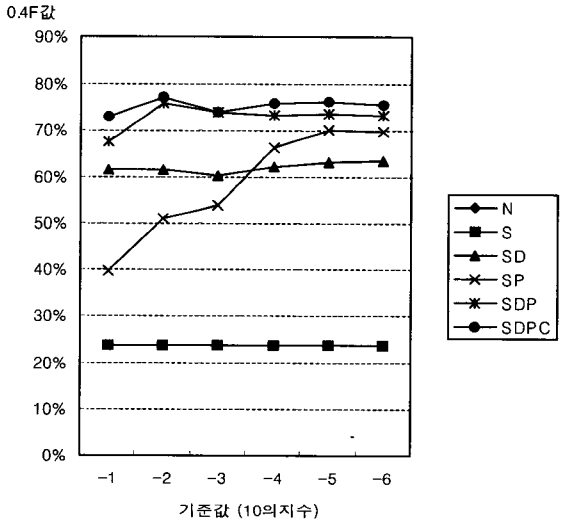
- EM(Exact Match) : 정확하게 정답과 일치된 추출결과의 수
- NE(No Extract) : 정답이 있으나 전혀 추출하지 않은 정답의 수
- FE(False Extract) : 정답이 없는데도 있는 것으로 알고 잘못 추출한 결과의 수

대응 번역어 추출을 위해 단순 단어에 대한 대역 가능성을 검사하기 위한 실험으로는 1) 단어 사용형태 그대로 일치하는 방법(방법 N), 2) 원형 복원 후 완전 일치 방법(방법 S), 3) 원형 복원 및 사전 대역어의 부분 일치(방법 SD), 4) 원형 복원 및 음운의 유사도에 의한 일치(방법 SP), 5) 원형복원, 음운유사도 및 사전 대역어의 부분일치

〈표 4〉 추출기준값에 따른 각 방법의 성능(0.4F)

(기준은 추출기준값으로 10의 지수승 : -1은 10^{-1})

방법 / 기준	N	S	SD	SP	SDP	SDPC
-1	23.8%	23.8%	61.4%	39.8%	67.7%	73.0%
-2	23.8%	23.8%	61.4%	51.0%	**75.9%**	**77.3%**
-3	23.8%	23.8%	60.1%	53.8%	74.1%	74.1%
-4	23.8%	23.8%	62.0%	66.2%	73.2%	75.9%
-5	23.8%	23.8%	63.0%	**70.3%**	73.7%	76.3%
-6	23.8%	23.8%	**63.3%**	69.7%	73.4%	75.7%



(그림 3) 추출기준값에 따른 성능변화

를 모두 고려한 방법(방법 SDP), 6) SDP방법에 다시 복합어 처리 방법을 추가한 방법(방법 SDPC)으로 각각에 대해 실험하고 성능을 측정하였다. 이를 위해 일단 대응 번역 확률인 추출기준값을 10^{-1}에서부터 10^{-6}으로 변화시키며 각 방법의 0.4F값을 측정했으며, 그 결과가 <표 4>와 같다. 또, 이 표를 그래프로 표시한 것이 (그림 3)이다. 여기에서 N과 S방법은 완전 일치되는 것만을 추출하기 때문에 성능이 추출기준값에 관계없이 일정하게 표시되어 있으며, 우연히도 두 가지의 성능이 같게 나타났다. 대체적으로 각 방법의 성능을 보면, N, S, SD, SP, SDP, SDPC순으로 성능이 높아짐을 알 수 있다. 즉, 음운 유사도를 이용한 일치와 사전 대역어 부분일치가 모두 효과적이며, 추출기준값을 10^{-4}이하로 할 경우, 음운 유사도 일치가 사전 부분일치 방법보다 좀더 효율적이다. 또, 각 방법을 혼합하여 사용한 방법이 효과적이어서, 원형복원, 사전 부분일치, 음운 유사도, 복합어 처리를 모두 혼합하여 처리한 방법이 가장 효과적임을 보여준다.

6. 토 론

6.1 사전번역

대역어 사전에 나오는 영어와 대역어가 그대로 문서에서 쓰이는 경우는 매우 적다는 것을 실험을 통해 알 수 있다. (물론 대역어 사전의 규모와 정확도에 따라 성능이 매우 달라 질 수는 있지만, 새로운 단어나 새로운 번역어 등을 완벽하게 지원해 주는 대역어 사전이 없는 한 이러한 방법은 유효하다.) 즉, N방법은 0.4F값이 23.8%로 매우 낮아, 적절한 괄호 대역어를 추출하지 못했다. 또, 원형을 복원한 방법인 방법 S는 이번 실험에서 우연히도 성능에 전혀 기여하지 못해 방법 N과 똑같은 성능을 나타냈다. 이는 변형된 단어가 괄호 안의 단어에서 적게 사용되었거나, 원형복원이 완전일치 방법을 통해서는 괄호 대역어 추출에 기여하지 못했음을 보여준다.

사전 대역어의 부분일치를 사용한 SD방법은 성능이 많이 향상되어 0.4F값이 최고 63.3%를 기록하였다. 즉, 사전 대역어가 문서에서 사용된 대역어들과 차이가 있음을 보여준다. 실제 실험에 의해 나타난 예를 보면, "validation"이 문서에는 "확인"으로 나왔지만, 사전에는 보다 서술적인 형태로 "정당함을 인정함", "확인함" 등으로 나와 일치가 되지 않았다. 또, "query"에 대한 대역어로 문서에서는 "질의"가 사용되었지만, 사전에는 "질문", "의문", "물음표", "조회" 등으로만 표기되어 있어 일치가 되지 않았다. 이때 부분 일치를 통해 "확인"과 "질의"가 "validation"과 "query"의 대역어일 가능성을 확률로 계산하여 줌으로써, 대역어 추출이 가능했다. 물론 이러한 유사성은 시소러스 등을 이용하여 어느 정도 해결할 수도 있지만, 시소러스 구축의 어려움도 있으며, 그 유사도 계산 방법도 일정치 않아 문제가 있을 수 있다.

6.2 음차번역

음운 유사도를 이용한 일치 방법은 미등록어로 나온 단어들을 음차표기되었다고 가정하고 유사도를 비교하여 대역어를 추정하는 방법이다. 기본적으로 원형복원을 하여 사전 검색을 완전 일치로 한 후, 다시 음운 유사도로 계산하여 그 유사성이 높으면 음차표기된 대역어로 판단하여 괄호 대역어를 찾아낸다. 음차표기를 사용하여 새로 찾아낸 예를 보면, "휴우리스틱(heuristic)", "로케이션 트랜스페어런시(location transparency)", "메쉬(mesh)" 등이 있다. 실험 결과 S방법보다 0.4F값이 23.8%에서 최고 70.3%까지 증가하여 매우 효과적임을 알 수 있다.

6.3 종합방법, 복합어 처리 및 파라미터 조정

사전대역어 부분일치 방법과 음운 유사도에 의한 일치 방법을 모두 사용한 SDP방법은 두 가지 방법에 나타난 효과가 반영되어 나타났다. 실험 결과 S방법보다 0.4F값이 23.8%에서 최고 75.9%로 향상되어 증가 비율은 약 3.19배 (순수 증가된 값은 52.1%)로 매우 효과적임을 알 수 있다. 또, 복합어 처리를 한 경우, 이보다 더 증가하여 77.3%로 S방법보다 최고 3.25배(순수 증가된 값은 53.5%) 증가하였다. SDP방법에서 못 찾은 것을 SDPC방법이 찾아낸 예를 보면, "헤드리터럴(head literal)"이나 "작업메모리(working memory)"등이 있어 실제 추출에 기여하고 있음을 보여준다.

<표 4> 및 (그림 3)에서 보는 바와 같이 추출기준값이 각기 다른 부분에서 각 방법들의 최고값을 나타내고 있다. 따라서, 파라미터값을 다르게 적용할 경우, 실험에 나타난 결과보다도 성능이 향상될 가능성이 있다. 예를 들어, 사전번역의 추출기준과 음차번역 추출기준을 독립적으로 적용하거나, 사전내의 번역어 일치 정도 계산 방법을 변경시킴으로써, 성능향상을 시킬 수 있을 것이다.

6.4 오류 원인 분석

잘못된 부분을 분석해보면 크게 6가지로 분류하여 볼 수 있다.

1) 원문문제 : 원문 자체의 오류나 원문 자체가 대응어 추출 혼동을 충분히 줄 수 있는 경우이다. 즉, 원문에서 하이픈을 사용하여 단어가 분리됨으로서 사전 검색을 불가능하게 한 "divide-and-con-quer(분할정복)"과 철자오류 "assigment (정확한 단어는 assignment)" 같은 예를 들 수 있다. 또한 "Internet에 있는 한 집단(multicast group)"처럼 괄호 안의 단어가 대역어와 유사하면서 앞에서 설명한 내용을 다른 용어로 재정의한 형식이 있다. 이 경우, "multicast group"은 괄호

대역어로 볼 수 없으나, "group"과 "집단"이 사전일치를 통해 괄호 대역어로 해석되어 잘못된 결과를 추출하였다.

2) 미등록어 : 있어야 할 영어단어가 사전에 없기 때문에 사전일치가 이루어지지 않아 잘못 추출하는 경우이다. 이 경우는 일반 명사 혹은 동사/형용사 등이지만 사전에 없어서 일치가 안된 경우(예 : approximate, compacted)로서 엉뚱한 한글 단어에 음운유사도가 잘못 적용되어 추출되는 경우이다.

3) 음차 번역 오류 : 일반적으로 사전에 나타나지 않는 약어들이 특별한 설명없이 사용되어 혼동을 야기하는 경우(시스템 기술 관리자 : SDM, 전자통신연구소 : ETRI 등)가 있다. 이 경우도 마찬가지로 매우 낮은 확률로 음운유사도가 계산되어 잘못 추출되는 경우이다.

4) 대역어 부분일치오류 : 사전에 단어가 존재하지만 그 단어의 대역어가 부분일치에 사용됨으로써 잘못된 결과를 추출한 경우이다. 예를 들어 "initial"이라는 단어가 사전에 "머리글자"로 대역어를 가지고 있고, 이 단어가 "알고리즘"과 "리"자를 공유함으로써 부분일치가 계산되어 잘못 추출하는 경우이다. 또, "두 개의 O(n log n)"과 같은 문장에서 하나의 표기를 괄호 대역어 사용으로 잘못 인식하여 "n"이 "두 개의"와 일치되는 경우이다. 참고로 "n"은 사전에 (그림 4)와 같이 "북부 여러 주의 사람"이라는 대역어가 포함되어 공통적으로 "의"라는 음절을 포함하게 된다. (괄호 대역어 관계에서 영어 한글자는 대개 사용되지 않으므로, 기본 대역어 사전에서 이런 표제어를 제거함으로써 이런 오류를 줄일 수도 있을 것이다.)

n ; 굴절률 ; … 뉴턴 ; 북극 ; 북풍 ; 북국 사람 ; 북부 사람 ; 북부 여러 주의 사람 ; …

(그림 4) 표제어 "n"에 대한 사전 기술예

5) 대역어 부재 : 영어 단어는 사전에 존재하나 대역어가 실제 문서에서 사용된 것과 전혀 다를 경우이다. 예를 들어 "testing tool"이 "구조적 검사도구"로 잘못 추출되었는데, "testing"은 사전에서 "시험", "실험", "테스트", "테스트하기" 로만 나와 있고 "검사"라는 대역어는 존재하지 않을 뿐만아니라, 음절상으로도 유사한 단어가 존재하지 않았다. 따라서, "testing"이 "구조적"이라는 엉뚱한 단어와 낮은 확률로 음운유사도를 가지고 일치되었다.

6) 복합어 오류 : 한국어 띄어쓰기와 영어 띄어쓰기의 차이에 따라서 대응이 잘못되는 경우이다. 앞의 6.3절 분석에서와 같이 이미 복합어 처리를 하여 추출한 경우도 있지만, 좀더 복잡하게 복합어 관계로 처리된 단어들은 제대로 추출하지 못했다. 예를 들어 "부분 집합의 합(sum of subset)"은 (그림 2)에 나타낸 것 같이 복잡하게 여러 단어에 걸쳐 정렬되어 있어서, 제대로 추출하지 못했다. 이를 해결하기 위해서는 복잡한 복합어 관계를 계산하기 위한 방법이 더 연구되어야 한다.

오류는 위에서 분석한 여러가지 원인이 복합적으로 나타날 수도 있다. 분석의 편의를 위해 오류의 원인을 대표적인 것으로만 한정하여 분석하였으며, 그 결과는 <표 5>에 나타나 있다. 원인이 음차표기 잘못인 경우가 39.4%로 상당히 많은 부분을 차지하고 있으며, 두 번째로 대역어 부분일치가 잘못되어 발생되는 오류가 21.2%로 나타났고, 그 다음이 사전에 미등록된 단어 때문에 음차표기가 잘못되는 오류 및 원문에 철자오류 등의 문제로 발생되는 오류가 15.2%를, 유사한 대역어가 사전에 전혀 없어서 추출하지 못한 오류가 6.1%, 다중어 처리가 안되어 나타나는 오류가 3.0%를 각각 차지했다. 결과적으로 음차번역 오류와 대역어 부분일치 오류가 성능향상에 많은 도움을 줌과 동시에 많은 오류의 원인이 되기도 했다.

〈표 5〉 오류분석

(소수점 둘째자리에서 반올림)

원 인	비 율
음차오류	39.4 %
대역어 부분일치 오류	21.2 %
미등록어	15.2 %
원문오류	15.2 %
대역어 부재	6.1 %
다중어 처리	3.0 %
	100.0 %

7. 결 론

본 논문에서는 괄호와 함께 사용된 대역어구를 자동인식하기 위한 방법으로 사전에 기반한 대역어 부분 일치 방법, 미등록어 처리를 위해 음차 표기에 의한 일치방법 및 복합어 처리 방법을 제안하였다. 실험결과 이들 방법은 모두 대역어 인식 성능을 크게 향상시켜, F값(α가 0.4일 경우)을 기준으로 비교해 보면 이들 방법을 사용하지 않았을 때 23.8%에서 77.3%로 무려 3.25배의 성능향상을 보였다. 따라서, 논문에서 제안한 방법들인 대역어 부분일치방법, 음차표기를 이용한 일치방법, 복합어 처리방법이 매우 효과적임을 알 수 있다. 이러한 결과도 좀더 세밀하게 파라미터 값을 조절할 경우, 더 좋은 성능를 나타낼 수도 있을 것이다.

논문에서 제안된 방법들은 괄호와 함께 사용된 대역어 인식뿐만 아니라, 이중언어 코퍼스에서의 대역어 추출이나 용어일치 점검, 단어 유사도를 이용한 시소러스 구축 등에도 사용될 수 있을 것이다.

참 고 문 헌

[1] D. A. Hull and G. Grefenstette, "Querying across languages : a dictionary-based approach to multilingual information retrieval," in *Proceedings of ACM SIGIR Conference on Information Retrieval*, Zurich, Switzerland, pp.49-57, 1996.

[2] P. F. Brown, J. C. Lai, and R. L. Mercer, "Aligning sentences in parallel corpora," In *Proceedings 29th annual meeting of the ACL*, Berkeley, CA, pp.169-176, 1991.

[3] P. F. Brown and et al, The mathematics of statistical machine translation : parameter estimation, *Computational Linguistics*, Vol.19, No.2, pp.263-311, 1993.

[4] I. Dagan, K. Church, and W. Gale, "Robust bilingual word alignment for machine aided translation," In *Proceedings of the Workshop on Very Large Corpora : Academic and Industrial Perspectives*, pp.1-8, 1993.

[5] D. Wu and X. Xia, "Learning an English-Chinese lexicon from a parallel corpus," Association for Machine Translation in the Americas, Columbia, MD, pp.206-213, 1994.

[6] K. Church, "Char_align : A program for aligning parallel texts at the character level," in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Ohio, pp.1-8, 1993.

[7] D. Wu and P. Fung, "Improving Chinese tokenization with linguistic filters on statistical lexical acquisition," Fourth Conference on Applied Natural Language Processing, Stuttgart, pp.180-181, 1994.

[8] 신중호, "한국어/영어 병렬 코퍼스에 대한 단어단위 및 구단위 정렬모델", 석사학위논문, 한국과학기술원, 1996.

[9] 이재성, 다국어 정보검색을 위한 영-한 음차 표기 및 복원 모델, 박사학위논문, 한국과학기술원, 1999.

[10] 이재성, "번역문에서의 외래어 표기용례 자동구축", 컴퓨터정보통신연구, 9권 2호, 충북대학교 컴퓨터정보통신연구소, pp. 25-33, 2001.

[11] 박영찬, 최기선, 김재군, 김영환, "한국어 정보 검색 연구를 위한 시험용 데이터 모음 2.0(KTSET 2.0) 개발", 한국정보과학회 인공지능연구회 춘계학술발표대회논문집, 서울, pp.59-65, 1996.

이 재 성

e-mail : jasonl@cbu.ac.kr

1983년 서울대학교 컴퓨터공학과(학사)

1985년 한국과학기술원 전산학과(석사)

1999년 한국과학기술원 전산학과(박사)

1985년~1988년 큐닉스컴퓨터 개발부 과장

1988년~1989년 미국 마이크로소프트 S/W 설계자

1988년~1993년 마이크로소프트 개발부 차장

1999년~2000년 한국전자통신연구원 선임연구원/팀장

2000년~현재 충북대학교 컴퓨터교육과 전임강사

관심분야 : 정보검색, 자연언어 처리, 한글공학, 컴퓨터교육

서 영 훈

e-mail : yhseo@chungbuk.ac.kr

1983년 서울대학교 컴퓨터공학과(학사)

1985년 서울대학교 컴퓨터공학과(석사)

1991년 서울대학교 컴퓨터공학과(박사)

1988년~현재 충북대학교 컴퓨터공학과 교수

1994년~1995년 미국 Carnegie-Mellon 대학 기계번역센터 객원교수

관심분야 : 자연언어분석, 음성언어처리, 정보검색·분류 및 요약