김일성종합대학학보
정보과학

주체109(2020)년
제66권 제1호

코퍼스에서 단어의 출현빈도에 의한 꼬리표작성과 그것을 리용한 실례기지탐색의 한가지 방법

김 동 수

선행한 실례토대기계번역들[1－5]에서는 실마리어에 의한 류사성탐색, 어휘의 품사정보와 의미정보에 의한 류사성탐색, 어휘의 일치에 의한 류사성탐색을 진행하여 입력문에 류사한 실례문을 선택하고 선택된 목적문을 변형하여 번역문을 생성하고있다. 이러한 실례토대기계번역에서는 입력문에 대한 류사한 실례문장의 탐색을 실례기지에 있는 문장의 어휘들과의 비교로 진행하므로 실례기지탐색이 신속정확히 진행되지 못하고있다.

론문에서는 선행연구의 이러한 결함을 분석한데 기초하여 두 나라 언어로 된 병렬코퍼스의 원천문들에서 단어출현빈도와 매 단어가 속한 문장의 순서번호를 추출하여 꼬리표를 구축하고 이 꼬리표를 리용하여 입력문에 가장 류사한 원천문을 탐색하고 번역문을 출력하는 방법에 대하여 제안하였다.

1. 실례토대기계번역

실례토대기계번역(EBMT)은 이미 번역된 번역쌍들을 실례기지로 구축하고 입력문과 류사한 원천문을 포함한 실례기지의 실례문을 선택하여 입력문의 번역조건에 따라 목적문을 변형하여 번역을 진행하는 체계이다.

EBMT는 입력문과 류사한 실례기지의 원천문을 탐색하기 위하여 여러가지 방법을 리용하고있다.

실마리어에 의한 EBMT[1]는 문장의 기본사상을 반영하는 실마리어를 실례기지의 탐색실마리어로 선정하고 입력문과 실례문사이의 류사성탐색을 문장들의 실마리어의 모임론적비교로 진행하여 번역을 진행하는 체계이다. 이 체계는 대화번역에 많이 리용된다.

어휘의 품사정보와 의미정보에 의한 EBMT[2, 4]는 입력문과 실례문사이의 류사성탐색을 먼저 품사정보에 의한 구조류사도탐색을 진행하여 문장론적으로 류사한 실례문들을 선택하고 다음 의미정보에 의한 의미류사도탐색을 진행하여 입력문에 가장 적합한 실례문을 선택하여 번역의 다의성문제도 해결하는 체계이다.

어휘의 일치에 의한 EBMT[3]는 문장을 이루는 어휘들의 일치에 의한 류사성탐색을 진행하여 선택된 실례기지의 목적문에서 일치한 어휘에 대한 번역은 그대로, 일치하지 않는 어휘에 대한 번역은 그 어휘의 번역을 목적문의 어휘와 치환하여 입력문에 대한 번역을 진행하는 체계이다. 그리고 번역형타를 리용하여 련결하는 어휘쌍들의 치환을 진행하여 번역을 진행하는 EBMT와 구단위패턴을 리용하여 부분번역을 진행하는 EBMT[5]도 있다.

이러한 EBMT들은 대부분 입력문과 실례기지의 원천문사이의 어휘와 어휘정보들의 비교에 의하여 류사성탐색을 진행하므로 입력문에 류사한 원천문의 신속정확한 탐색을 보장하지 못한다.

론문에서는 EBMT체계에서 중요한 입력문에 가장 류사한 원천문을 신속정확히 찾을

수 있도록 병렬코퍼스에서 단어의 출현빈도에 따르는 꼬리표작성과 그것을 리용한 실례기지탐색방법에 대하여 제안하였다.

2. 단어출현빈도에 따르는 꼬리표작성

실례토대기계번역의 정확성과 유연성을 보장하려면 번역하려는 입력문장과 동등하거나 류사한 병렬코퍼스의 원천문장을 신속정확히 탐색하는것이 중요하다. 입력문장과 병렬코퍼스의 원천문장과의 류사성탐색을 어떤 방법에 의하여 진행하는가에 따라 번역체계의 신속정확성이 결정되게 된다.

여기서는 병렬코퍼스의 원천문장들에 있는 단어들의 출현빈도수와 빈도수에 따르는 문장들의 순서번호들로 꼬리표를 만들어 리용하는 방법을 고찰한다.

코퍼스에 있는 매 단어들의 출현빈도수와 단어를 포함하는 문장의 코퍼스에서의 순서번호는 코퍼스의 원천문장들의 매 단어들의 비교로서 얻는데 다음과 같이 진행한다.

① 병렬코퍼스의 원천문장들로 된 코퍼스에서 첫 문장을 읽는다. 이 문장을 형태소해석하여 토큰들로 분리한다. 여기서 토큰은 분리된 단어의 형태소해석결과이다.

② 분리된 토큰들에 대한 정보를 모두 출현수는 1, 문장번호모임은 원소 {1}로 설정한다.

③ 두번째 문장을 형태소해석하여 토큰들로 분리한다.

첫 문장의 첫번째 토큰에 대하여 두번째 문장의 토큰들과 비교하여 일치하는것이 있으면 그 토큰에 대한 정보를 갱신한다. 즉 출현수는 2, 문장번호모임은 {1, 2}로 갱신한다.

④ 두번째 토큰에 대하여 두번째 문장의 토큰들과 비교하여 일치하는것이 있으면 정보를 첫 토큰과 같이 갱신하며 계속하여 나머지토큰들에 대하여 비교를 진행한다.

이와 같이하여 첫번째 문장과 두번째 문장의 토큰들의 비교가 끝나면 첫번째 문장에 있는 토큰들의 정보가 갱신된다.

⑤ 세번째 문장의 토큰들에 대하여 우에서 진행한 방법으로 일치비교를 진행하고 일치하는 토큰이 있으면 토큰들의 정보를 갱신한다.

이와 같은 방법으로 코퍼스의 마지막문장까지 일치비교를 진행하여 토큰들의 정보를 갱신한다.

그러면 병렬코퍼스의 원천문장들에 있는 모든 토큰들에 대한 출현빈도수와 출현문장번호모임으로 구성된 정보가 얻어진다. 이 정보가 붙은 매 토큰들을 자모순에 따라 정돈하여 실례토대기계번역을 위한 꼬리표를 만든다.

3. 꼬리표작성 및 갱신알고리듬

실례기지에 있는 원천문장의 토큰을 ew_i, 입력문장을 토큰화한 결과를

$$s_i = \{ew_{i1}, ew_{i2}, \cdots, ew_{in}\}$$

으로, ew_{ij}의 출현빈도수를 Cn_{ij}, 문장순서번호모임을

$$Sn_{ij} = \{index_1^i, index_2^i, index_3^i, \cdots, index_t^i\}$$

로 표시하자. 여기서 $index_j^i$는 코퍼스에서 문장의 순서번호이다.

꼬리표작성알고리듬은 다음과 같다.

입력 : 실례기지문장들

출력 : ew_i의 Cn_i(빈도수), Sn(문장순서번호모임)

① $i = 1$

② S_i선택, 토큰화 $ew_{i1}, ew_{i2}, \cdots, ew_{in}$

$i = 1$인 경우 $Cn_{1j} = 1$, $Sn_{1j} = \{1\}(j = 1, \cdots, n)$

$i \neq 1$인 경우 $ew_{ij} \neq ew_{lkl}$인 ew_{ij}에 대하여 $Cn_{ij} = 1, Sn_{ij} = \{1\}$로 설정

여기서 $l = 1, 2, \cdots, i-1, kl = 1, 2, \cdots, n_{i-1}, j = 1, 2, \cdots, n_i(n_i : i$문장의 단어수)이다.

③ $j = i+1$

④ S_j를 선택, 토큰화 $ew_{j1}, ew_{j2}, \cdots, ew_{jm}$

⑤ $ew_{il} = ew_{jk}$ 비교($l = 1, \cdots, n, k = 1, \cdots, m$)

ㄱ) 성립 ⇒ $ew_{il} : Cn_{ij} = Cn_{ij} + 1, Sn_{ij} = Sn_{ij} \cup \{j\}$

ㄴ) 비성립⇒ Cn_{ij}, Sn_{ij}는 그대로

⑥ $j < n$이면 $j = j + 1$로 놓고 ④에로 이행

⑦ $i = i + 1, i < n$이면 ②에로 이행

⑧ ew_{ij}를 ew_i로 표시하고 그것에 대응하는 빈도수와 문장번호를 Cn_i, Sn_i로 표시

⑨ ew_i들을 자모순으로 정돈

⑩ 꼬리표작성

꼬리표를 형식화하면 다음과 같다.

토큰을 ew_i, 코퍼스에 있는 토큰들의 모임을 EW라고 하면

$$EW = \{ew_i, i = 1, 2, \cdots, N\}$$

이다. 여기서 N은 코퍼스의 원천문장들에 있는 토큰들의 총개수이다.

매 토큰에 대한 정보는 다음과 같이 표시된다.

$$ew_i(Cn_i, Sn_i)$$

실례토대기계번역에 리용될 정보가 있는 토큰모임을 EWT라고 하면

$$EWT = \{ew_1(Cn_1, Sn_1), ew_2(Cn_2, Sn_2), \cdots, ew_N(Cn_N, Sn_N)\}$$

이다.

EWT에 있는 토큰들을 자모순으로 정돈하여 토큰과 토큰의 빈도수, 문장출현번호모임으로 구성된 꼬리표를 작성한다.

꼬리표의 갱신은 즉 새로운 번역문쌍이 주어지면 원천문의 토큰들을 꼬리표에서 검색하고 꼬리표에 있는 토큰들에 대하여서는 Cn_i, Sn_i를 갱신하며 없는 토큰들에 대해서는 우와 같은 방법으로 꼬리표정보를 결정하고 꼬리표에 자모순관계를 고려하여 추가한다.

꼬리표갱신알고리듬은 다음과 같다.

① S_{n+1}: 토큰화 $ew_{n+1\,1}, ew_{n+1\,2}, \cdots, ew_{n+1\,t}$

② $ew_{n+1\,i}$를 꼬리표에서 탐색

ㄱ) 있으면 $Cn_i + 1$, $Sn \cup \{n+1\}$

ㄴ) 없으면 꼬리표작성알고리듬에 의해 Cn_{N+1}과 Sn_{N+1}을 결정

③ $ew_{n+1\,i}(Cn_{N+1}, Sn_{N+1})$: 꼬리표에 자모순에 따라 추가

병렬코퍼스에서 122개의 영어문장과 번역된 조선어문장에 대한 꼬리표를 작성하면 표와 같다. 이 표는 122개의 영어문장에 있는 단어수 1 080개 가운데서 20개 단어들의 출현수와 문장순서번호모임으로 된 꼬리표이다.

표. 병렬코퍼스에서 122개의 영어문장과 번역된 조선어문장에 대한 꼬리표

번호	단어	출현수	문장번호
1	addition	2	2, 10
2	amount	3	2, 48, 77
3	are	10	1, 6, 8, 10, 13, 23, 72, 73, 86, 89
4	constant	4	2, 78, 109
5	copper	2	2, 102
6	experiment	4	3, 15, 68, 84
7	for	8	2, 63, 65, 67, 71, 92, 100, 102
8	hardene	1	2
9	he	3	3, 104, 109
10	his	1	3
11	last	1	3
12	microscope	1	1
13	molecule	2	1, 10
14	most	3	1, 6, 13
15	neglect	1	3
16	person	1	3
17	powerful	1	1
18	pure	1	1
19	seen	1	1
20	silver	1	2
21	small	3	1, 3, 37
22	soft	1	2
23	too	4	1, 2, 67
24	use	5	2, 76, 102, 121, 122
25	usually	1	2
26	with	12	1, 11, 50, 54, 87, 101, 109, 110, 111, 112, 114, 115

4. 꼬리표를 리용한 실례기지탐색

실례토대기계번역은 입력문과 실례기지의 원천문과의 류사성탐색을 전제로 하는데 류사성탐색은 여러가지로 진행된다.

여기서는 작성된 꼬리표를 리용하여 입력문장과 실례기지의 원천문과의 류사성탐색을 진행하는 방법에 대하여 취급하였다.

먼저 입력문장을 토큰들로 분리한다.

다음 매 토큰에 대하여 꼬리표에 있는 토큰정보(Cn_i, Sn_i)를 얻는다.

이 토큰정보들을 리용하여 입력문에 류사한 실례기지의 원천문탐색을 진행하는데 방법은 다음과 같다.

입력문장을 IS라고 하고 IS의 토큰들은 iew_i로 표시하면 입력문은 다음과 같이 표시된다.

$$IS = \{iew_1, iew_2, iew_3, iew_4, \cdots, iew_m\}$$

여기서 m은 입력렬의 토큰의 개수이다.

다음 매 토큰의 꼬리표에서 정보를 표시하면 $iew_i(Cn_i, Sn_i)(i = 1, 2, 3, \cdots, m)$이다.

토큰들의 꼬리표정보에서 $Sn_i(i = 1, 2, \cdots, m)$들을 선택하면 다음과 같이 표시된다.

$$Sn_1 = \{index_1^1, index_2^1, \cdots, index_{s1}^1\}$$
$$Sn_2 = \{index_1^2, index_2^2, \cdots, index_{s2}^2\}$$
$$Sn_3 = \{index_1^3, index_2^3, \cdots, index_{s3}^3\}$$
$$\vdots$$
$$Sn_m = \{index_1^m, index_2^m, \cdots, index_{sm}^m\}$$

다음 Sn_i 들의 문장순서번호모임의 합

$$SN = \bigcup_{i=1}^{m} Sn_i = \bigcup_{i=1}^{m} \bigcup_{j=1}^{sn_i} \{index_j^i\}$$

를 얻는다.

SN에서 출현수가 제일 높은 $index_j^i$ 를 선택하면 다음과 같다.

$$index_0 = \underset{freq(index_j^i)}{\arg\max} \{index_j^i,\ i = \overline{1,\ m}, j = \overline{1, sm}\}$$

웃식에 대응되는 실례기지의 원천문장들이 입력문에 가장 류사한 문장들로 되며 대응되는 목적문들이 입력문에 대한 번역문의 후보로 선택된다. 여기서 입력문과의 어휘비교를 진행하여 제일 류사한 문장을 번역후보로 선택한다.

선택된 원천문에 대응되는 목적문과 입력문사이의 단어대응관계에 따르는 변환 즉 I(삽입), R(치환), D(제거), N(직접출력)변환을 실현하여 번역문을 생성한다. 꼬리표를 작성하고 입력문의 토큰들의 꼬리표에서의 문장출현정보를 리용하여 실례기지탐색을 진행한다.

맺 는 말

선행한 EBMT방법들을 연구한데 기초하여 코퍼스에 있는 단어들의 출현빈도수와 그 단어가 출현한 수만 한 문장순서번호들로 구성된 꼬리표를 작성하고 입력문과 실례기지와 류사성탐색을 어휘의 비교가 아니라 꼬리표에서 단어탐색으로 진행하여 실례토대기계번역에서 입력문과 실례기지의 류사성탐색속도와 정확도를 높이였다. 이 방법에서는 코퍼스량이 클수록 어휘탐색에 비해 효과가 더 높아진다.

참 고 문 헌

[1] 김동수; 전자공학, 3, 42, 주체88(1999).

[2] 김동수; 정보과학과 기술, 6, 25, 주체92(2003).

[3] **김일성**종합대학학보(자연과학), 53, 1, 52, 주체96(2007).

[4] Eiji Aramaki, Sadao Kurohashi; International Workshop on Spoken Language Translation, 91, 2004.

[5] Liling Tan et al.; In Proceedings of the 9th International Workshop on Semantic Evaluation, 85, 2015.

주체108(2019)년 11월 5일 원고접수

A Method of Making of Tag Table with Appearance Frequency of Word in Corpus and Example Base Search Using It

Kim Tong Su

The paper proposes a method of machine translation by using appearance frequency of word and order number of sentence having each word in source sentences of parallel corpus.

Keywords: EBMT, parallel corpus, word matching