한영 기계번역에서의 효율적인 구문분석과 번역을 위한 유한상태 변환기 기반 전처리기의 설계 및 구현

박준식, 최기선
전문용어언어공학연구센터
한국과학기술원 전산학과
대전시 유성구 구성동 373-1, 우:305-701
{jspark, kschoi}@world.kaist.ac.kr

Design and Implementation of Finite-State-Transducer Preprocessor for an Efficient Parsing and Translation in Korean-to-English Machine Translation

Junsik Park, Key-Sun Choi
Korea Terminology Research Center for Language and Knowledge Engineering
Department of Computer Science
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, 305-701

요약

기계번역이나 정보검색 등에 적용되는 자연언어처리기술에 있어서 구문분석은 매우 중요한 위치를 차지한다. 하지만, 문장의 길이가 증가함에 따라 구문분석의 복잡도는 크게 증가하게 된다. 이를 해결하기 위한 많은 노력 중에서 전처리기의 지원을 통해 구문분석기의 부담을 줄이려는 방법이 있다.

본 논문에서는 구문분석의 애매성과 복잡성을 감소시키기 위해 유한상태 변환기 (Finite-State-Transducer; FST)를 이용한 전처리기를 제안한다. 유한상태 변환기는 사전표현, 단어분할, 품사태깅 등에 널리 사용되어 왔는데, 본 논문에서는 유한상태 변환기를 이용하여 형태소 분석된 문장에서 시간표현 등의 제한된 표현들을 구문요소화하는 전처리기를 설계 및 구현하였다. 본 논문에서는 기계번역기에서의 구문분석기 뿐만 아니라 변환지식의 모듈화를 지원하기 위해 유한상태 변환기를 이용하여 시간표현 등의 부분적인 표현들을 번역하는 방법을 제안한다. 또한 유한상태 변환기의 편리한 작성을 위하여 유한상태 변환기 작성 지원도구를 구현하였다. 본 논문에서는 전처리기의 적용을 통해 구문분석기의 부담을 덜어 주며 기계번역기의 변환부분의 일부를 성공적으로 담당할 수 있음을 보여 준다.

1 서론

기계번역이나 정보검색 등에 적용되는 자연언어처리기술에 있어서 구문분석은 아주 중요한 위치를 차지한다. 하지만, 문장의 길이가 증가함에 따라 구문분석의 복잡도는 크게 증가하게 된다[2,3]. 이를 해결하기 위한 많은 노력 중에서 전처리기의 지원을 통해 구문분석기의 부담을 줄이려는 방법이 있다[3]. 본 논문에서는 기계번역 시스템_하에서 형태소 분석과 구문분석과 변환 사이의 고정된 표현들을 부사구와 같은 구문요소 단위로 묶어서 구문분석과 변환의 효율을 높이는 전처리기 시스템을 제안한다. 제안하는 전처리기 시스템은 고정된 표현만을 다루며, 그 번역도 제한된 어휘와 의미를 다루기 때문에 유한상태 변환기 (Finite State Transducer)를 기반으로 하며,

작성을 쉽게 하기 위한 인터페이스와 간단한 시뮬레이션 환경을 제공한다.

유한상태 (FS; Finite-State) 모델은 최근 들어 기계번역에 이용되는 사례가 나타나고 있는데, 초기에는 기계번역과 같은 복잡한 문제에 FS 모델을 적용하는 것은 너무 간단한 것이 아닌가 하는 우려가 있었지만, 제한된 영역에서의 실제 적용해본 결과, 기대이상의 좋은 성과를 보여주었다[6,11]. 그 이유는 비교적 적은 양의 어휘와 의미 영역에 대해서 적용하였고, 또한 적용예가 언어적인 유사성이 높은 유럽언어 간이어서 각 언어간의 번역을 위해 정의된 매핑이 비슷하기 때문이었다.

하지만 한국어와 영어와 같이 그 언어적 구조가 다른 경우에는 단순한 FS 모델을 적용하는 것은 여러 가지 어려움을 가지고 있기 때문에, 그 적용에 대한 고려가 필요하다.

한영 기계번역의 경우, 한국어 형태소분석의 결과는 구문분석에 큰 영향을 미치며, 형태소분석 결과로부터 구문분석을 위해 구문요소를 형성하는 과정의 필요성이 제기되는데[3], 이렇게 형성된 구문요소는 비교적 제한된 어휘와 표현을 가지게 되어 번역에 있어서도 직접 번역 등의 방식을 이용할 수 있다. 본 논문에서는 시간표현, 주소 등의 번역과 같은 제한되어 있으며 고정적인 표현방식에의 적용을 시도한다.

본 논문의 구성은 다음과 같다. 2절에서는 기존의 유한상태 변환기를 이용한 번역 모델에 대해서 설명하고, 3절에서는 본 논문에서 제안하는 유한상태 변환기를 이용한 전처리기에 대해서 설명한다. 4절에서는 결론과 향후 연구사항에 관해 서술한다.

2 기존 연구

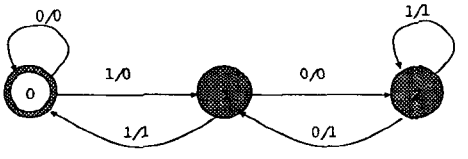
기계번역 시스템에 사용되는 번역 모델은 크게 제한성에 기반한 모델과 어휘에 기반한 모델로 나눌 수 있다. 어휘에 기반한 모델은 입력할 때, 고려할 필요가 있는 어휘에 관련되어 모델을 구성하게 되며, 강건하며 효율적인 장점을 가진다. 이러한 어휘에 기반한 모델 중에 유한상태 변환기 모델이 있다.

본 논문에서 제안하는 전처리기에서 사용하는 모델인 유한상태 변환기는 유한상태 오토마타 (Finite State Automata; FSA)의 전이함수의 값이 한 개의 기호로 이루어진 것에 비해, 기호의 쌍으로 이루어진 것이다. 엄밀한 정의를 하면 다음과 같다[9].

정의 (FST) 유한상태 변환기 T는 6개의 요소 $\left(\sum_1, \sum_2, Q, i, F, E\right)$로 이루어져 있으며, 이때 각 요소는 다음과 같다.

- $\sum_1$ 은 유한 개수의 입력 알파벳이다.
- $\sum_2$ 은 유한 개수의 출력 알파벳이다.
- Q는 상태 (state)의 유한 집합이다.
- $i \in Q$ 는 초기상태이다.
- $F \subseteq Q$ 는 최종상태의 집합이다.
- $E \subseteq Q \times \sum_1^* \times \sum_2^* \times Q$ 는 전이함수의 집합이다.

예를 들어 유한상태 변환기
$T = (\{0,1\},\{0,1\},\{0,1,2\},\{0\},\{0\},E)$
이며, 여기서
$E = \{(0,0,0,0),(0,1,0,1),(1,0,0,2),(1,1,1,0),(2,1,1,2),(2,0,1,1)\}$
일때, [그림 1]과 같이 나타낼 수 있다.



[그림 1] 유한상태 변환기

일반적으로 유한상태 변환기는 전자사전의 구조, 형태소 해석, 태거, 지역 문법 (local grammar), 정보 추출 등에 널리 이용되어 왔다[4,9].

기계번역을 위한 유한상태 변환기 모델에 대한 연구는 일반적인 유한상태 변환기 모델에 기반한 방식[10,11]과 헤드 오토마타 (head automata) 모델[6,7]이 있다.

2.1 기계번역을 위한 유한상태 변환기 모델[10]

부차적 연속 변환기 (Subsequential Transducer; SST)란 결정성 FS 네트웍으로 이루어져 있다. 입력 문장의 각각의 심볼에 대해 대응되는

문자열이 출력되며, 새로운 상태로 전이하게 된다. 모든 입력 심볼에 대해 처리가 끝나게 되어 최종 (final) 상태에 들어가게 되면, 추가로 출력이 생성되게 된다.
SST의 번역능력은 정확한 번역이 나올 수 있을 정도의 충분한 입력이 들어올 때까지 출력문의 생성을 "연기 (delay)"하는 것에서 비롯된다.

예를 들어, 다음의 문장

나는 학교에 간다. I go to school.

에 있어 한국어문장에서 영어문장으로의 번역은 다음과 같이 진행된다.
"나는"이라는 어절은 "I"라는 출력문을 생성하며, "학교에"라는 어절은 SST의 상태를 변경시키지만 출력문이 생성되지 않는다. 다음 어절 "간다"라는 어절은 "go"을 생성하며, 문장의 종결표시는 "to school."을 생성하게 된다. 이러한 유한상태 변환기를 추출하기 위한 연구도 많이 있는데 그 대표적인 것은 Onward Subsequential Transducer Inference Algorithm (OSTIA)[8]이다. OSTIA방법은 훈련세트를 일반화시키는 방식으로, 접두트리 (prefix-tree) 표현을 변환하게 된다.

2.2 헤드 오토마타 모델[6]

헤드 오토마타 모델은 어휘와 관련한 유한상태기계이며, 변환기의 경우 어휘쌍에 관련된 것이다. 헤드 오토마타는 관계형 헤드 오토마타 (relational head automata)와 헤드 변환기의 2개 오토마타로 이루어져 있다. 관계형 헤드 오토마타는 구문분석이나 생성 등에 사용되게 되며 헤드 변환기는 직접 번역 (direct translation) 등에 이용되는 모델이다. 본 절에서는 이들 헤드의 기본이 되는 "단순 헤드 오토마타" (Simple Head automata)에 대한 설명을 먼저 하고, 관계형 헤드 오토마타와 헤드 변환기에 대한 설명을 하고자 한다.

2.2.1 단순 헤드 오토마타

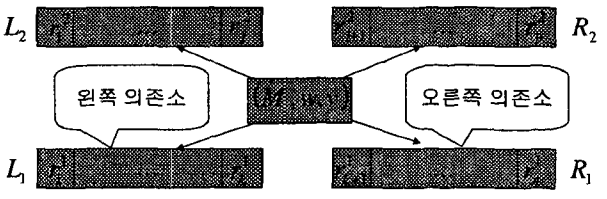
단순 헤드 오토마타의 정의는 다음과 같다. 어휘 V의 각각의 단어 w는 헤드 오토마타 M_w와 관련되어 있다. 헤드 오토마타 M_w는 상태의 집합 Q를 가지고 있으며, 그 중 시작 상태로 구분되는 q_0를 가진다. M_w는 V로부터의 단어의 연속된 열을 가진 쌍 (L, R)을 출력하게 된다. 여기서 L과 R은 단순 헤드 오토마타가 언어적 모델로 적용될 때, 각기 단어 w의 왼쪽과 오른쪽의 의존소 열을 의미하며, M_w는 L, w, R의 전체 문자열을 인식하게 된다. M_w는 좌전이 (left transition), 우전이 (right transition), 정지 등 3개의 행동을 취할 수 있다. 좌전이의 경우 M_w는 L의 오른쪽 끝에, 우전이의 경우 M_w는 R의 왼쪽 끝에 출력 단어를 쓰게 된다.

2.2.2 관계형 헤드 오토마타

관계형 헤드 오토마타는 단순 헤드 오토마타에 대해서 전이값에 사용된 기호가 의존 관계 기호 (dependency relation symbol)라는 점만 다르다. 관계형 헤드 오토마타는 이러한 관계 기호의 쌍을 인식하는 모델이라고 볼 수 있으며, 여기서의 관계는 헤드 단어와 헤드 단어의 좌우 주위에서 의존하는 구절들 간의 관계를 의미한다.

2.2.3 헤드 변환기

헤드 변환기는 양국어에서 동시에 나타나는 단어간의 공기관계에 대해 동시에 적용가능하게 된다. [그림 2]에서 헤드 변환기 M은 w의 왼쪽 의존소의 열 $\langle w_1 \dots w_k \rangle$과 오른쪽 의존소의 열 $\langle w_{k+1} \dots w_n \rangle$을 v의 왼쪽 의존소의 열 $\langle v_1 \dots v_j \rangle$과 오른쪽 의존소의 열 $\langle v_{j+1} \dots v_p \rangle$로 변환하게 된다. 이러한 헤드 변환기의 집합체를 이용하여 각 언어의 문자열을 변환하게 되며 이러한 변환 과정을 재귀적 헤드 변환기 (recursive head transducer)라고 부른다. 일반적인 유한상태 변환기의 경우에는 입력언어와 출력언어간의 순서가 임의적이면 해당하는 모델의 상태의 수가 증가하게 되지만, 재귀적 헤드 변환기에는 해당하지 않는다.



[그림 2] 헤드 변환기

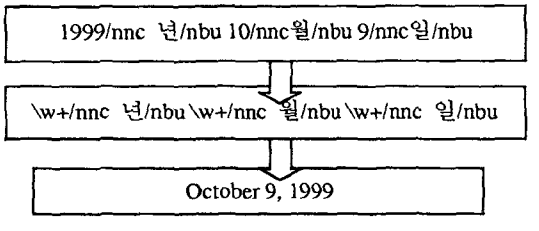
3 제안하는 모델

제안하는 전처리기는 날짜, 시간 및 단위가 있는 형태소의 열을 인식하여 대응하는 영어 표현으로 변환하게 된다. 이와 같은 구문요소는 다음과 같은 접사나 의존명사와 결합하게 된다[3].

- 순한글 양수사와 결합하는 단위성 의존명사: 마리, 필, 병, ...
- 한자어 양수사와 결합하는 단위성 의존명사: 년, 월, 원, ...
- 모든 양수사와 결합하는 단위성 의존명사: 척, 명, 평, ...

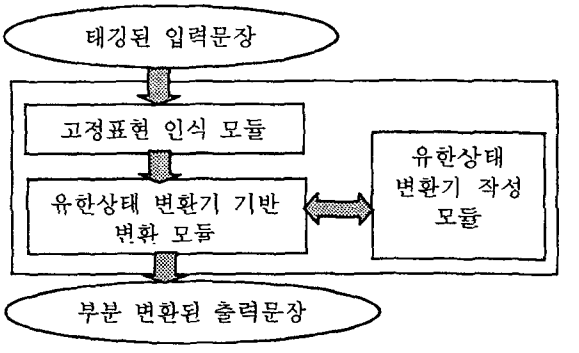
제안하는 전처리기의 모델에 부차적 연속 변환기를 도입하기에는 한영 표현상의 대역순서 차가 크기 때문에 곤란하여, 헤드 오토마타 모델을 기반으로 하였다. 하지만, 헤드 오토마타 모델은 좌우 의존소와의 의존확률을 기반으로 구문분석 및 변환을 하게 된다. 제안하는 전처리기는 사용자의 용이한 유한상태 변환기의 작성을 목표로 하였기 때문에 기본적인 유한상태 변환기 모델에 형태소를 단위로 하며, 헤드 오토마타 모델의 전이함수의 좌전이, 우전이 등을 도입하였다.

[그림 3]은 "1999년 10월 9일"의 한국어 표현을 "October 9, 1999"의 영어 표현으로 변환하는 예를 보여준다.



[그림 3] 시간표현의 변환[1]

제안하는 모델은 전체적으로 3개의 부분으로 나눌 수 있다. 고정표현 인식부분, 인식된 표현의 변환을 위한 변환부분, 인식과 변환 유한상태 변환기 작성을 위한 인터페이스 부분으로 나눌 수 있다.



[그림 4] 전처리기의 구조

3.1 고정표현 인식부분

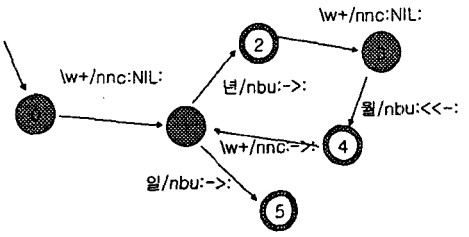
시간이나 장소에 관련한 주소표현 등을 정규식(regular expression)으로 나타내어서 실제 문장의 형태소분석된 결과에서 인식하게 된다. 본 논문에서 적용하는 분야는 무역분야 서신문 10,000여 문장에 대해서 주요한 시간 표현 등을 추출하였다. 다음은 해당 표현을 정규식으로 나타낸 것이다.

정규식	예
\w+/nnc 년/nbu \w+/nnc 월/nbu \w+/nnc 일/nbu	1991/nnc 년/nbu 1/nnc 월/nbu 1/nnc 일/nbu
\w+/nnc 월/nbu \w+/nnc 일/nbu 오후/ncn \w+/nnc 시/ncn	4/nnc 월/nbu 10/nnc 일/nbu 오후/ncn 2/nnc 시/ncn
모델/ncn+번호/ncn \w+/f -/sd \w+/nnc	모델/ncn+번호/ncn MI/f -/sd 500/nnc

[그림 5] 고정표현 규칙의 일부 및 그 예[2]

[1] 그림에서 \w+는 하나 이상의 문자를 나타내며, /nnc는 양수사, /nbu는 단위성 의존명사를 뜻한다[4].

[2] 정규식 문법은 perl5문법을 기준으로 하였다.



[그림 6] 시간표현을 위한 유한상태 변환기 중 일부

입력	현재상태	전이함수	현재출력
1999/nnc	0	\w/nnc:NIL:	
년/nbu	1	년/nbu:->:	1999
10/nnc	2	\w/nnc:NIL:	1999
월/nbu	3	월/nbu:<<-:	October 1999
9/nnc	4	\w/nnc:NIL:	October 1999
일/nbu	1	일/nbu:->:	October 9 1999
입력의 끝	5		October 9 1999

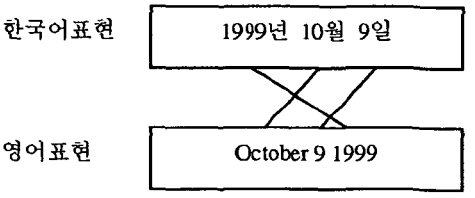
[그림 7] 시간표현의 유한상태 변환기를 이용한 변환과정

3.2 고정표현 변환부분

인식된 표현을 입력으로 유한상태 변환기를 통해 번역을 하게 된다. 출력되는 표현이 출력 문장의 현재 위치에서 어떻게 연결되는지 나타내는 것은 valency 연산자가 나타내며, 그 연산자는 다음의 5가지로 정의할 수 있다.

- ->: 현재 출력위치의 오른쪽으로 출력
- <-: 현재 출력위치의 왼쪽으로 출력
- ->> : 현재 출력문의 오른쪽 끝으로 출력 위치를 이동하여 출력
- <<- : 현재 출력문의 왼쪽 끝으로 출력 위치를 이동하여 출력
- NIL : 출력하지 않고 다음 입력과 합쳐진다.

예를 들어 "1999년 10월 9일" 입력표현에 대한 영어 표현은 "October 9 1999"의 형태로 나타낼 수 있다. 여기서 고려해야 할 점은 두_가지로 볼 수 있다. 첫번째는 "1999년"이 번역표현에서는 "1999"로 가장 뒤에 나타나게 된다. 두_번째는 "10월"이라는 단어가 "10+월"의 형태로 결합하여 "October"로 변환하게 되는 한국어와 영어간의 형태소 개수가 일치하지 않게 된다.



이와 같은 형태를 번역하기 위한 유한상태 변환기는 [그림 6]과 같다.

[그림 6]의 유한상태 변환기는 입력이 "1999년 10월 9일"인 경우, [그림 7]에 나타난 순서로 동작하게 된다.

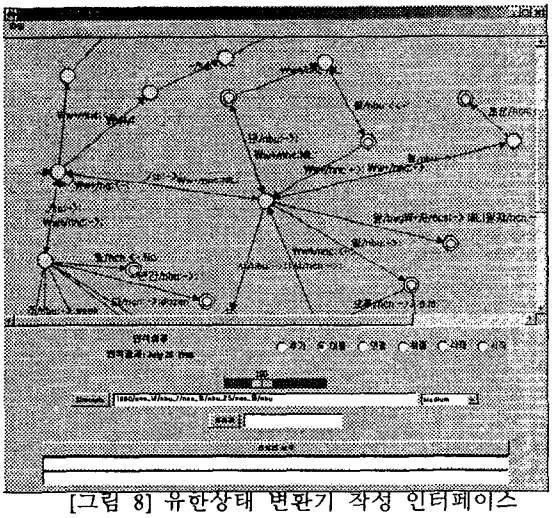
먼저 "1999/nnc"가 들어오면 초기상태는 상태 0이고, 해당하는 valency는 "NIL"이므로 출력을 보류하고 다음의 입력과 결합하게 된다. 상태 1로 전이한 후 다음의 입력이 "년/nbu"이므로, valency가 "->"으로 현재 출력 ""의 오른쪽으로 출력값을 낸다. 이 때 출력값은 앞의 입력값인 "1999/nnc"와 "년/nbu"이 결합한 "1999/nnc년/nbu"가 변환 모듈의 입력이 되어, "1999"가 대응 영어 출력값이 된다.

다음의 입력인 "10/nnc"는 "월/nbu"와 결합하며, 이때 valency가 "<<-"이므로, 현재출력 "1999"의 가장 왼쪽으로 출력위치가 이동하게 되어, "October 1999"의 출력값을 가지게 된다. 마지막으로 "9/nnc", "일/nbu"는 valency가 "->"이므로, 현재의 출력위치인 "October"의 오른쪽으로 변환값인 "9"가 나타나게 된다. 그 결과는 "October 9 1999"와 같이 된다.

전처리기상에서는 인식된 고정표현의 마지막까지 이르게 되면, 변환된 값을 출력하고, 다음 고정표현을 계속해서 찾게 된다.

3.3 유한상태 변환기 작성 인터페이스

유한상태 변환기 작성 인터페이스는 전처리기에 사용되는 유한상태 변환기를 사용자에게 익숙한 인터페이스를 이용하여 쉽게 작성할 수 있도록 한다. 작성 인터페이스는 사용되는 노드의 추가 및 삭제, 전이 함수의 작성 및 해당 함수값 추가 및 변경 인터페이스를 제공한다. 다음은 유한상태 변환기 작성 인터페이스의 모습이다



[그림 8] 유한상태 변환기 작성 인터페이스

4 결론

본 논문에서는 기계번역기에서의 구문분석기 뿐만 아니라 변환지식의 모듈화를 지원하기 위해 유한상태 변환기를 이용하여 시간표현 등의 부분적이지만 고정된 표현들을 번역하는 방법을 제안하였다. 제안한 전처리기는 한국어에서 관용화된 시간표현, 주소 등의 표현을 하나의 구문요소로 묶어서 구문분석기에 대해 입력형태소의 수를 줄여서 구문분석의 효율을 증가시켰다. 이 구문요소에 대해 비교적 제한된 어휘와 표현으로 유한상태 변환기에 기반한 직접번역 방식의 변환기로 표층적인 표현을 미리 번역하도록 하였다. 또한, 유한상태 변환기의 편리한 작성을 위하여 유한상태 변환기 작성 지원 인터페이스를 구현하였다. 본 논문에서는 이상과 같은 전처리기의 모델의 설계와 실제 적용을 통해 구문분석기의 부담을 덜어 주며 기계번역기의 변환부분의 일부를 담당함을 보여주었다.

현재, 전처리기는 현재 진행 중인 구단위 패턴기반 한영 기계번역 시스템[1]의 일부로 구단위 패턴매칭 부분에 대해 입력을 제공하는 역할을 위해 구현되었다. 구단위 패턴기반 한영 기계번역 시스템은 한국어 구문분석과 변환을 동시에 수행하는 특징을 가지고 있으며 본 전처리기는 구단위 패턴매칭 부문의 효율을 높이기 위한 구문요소화와 변환을 수행한다.

본 전처리기에서 사용한 유한상태 변환기는 형태소 해석과 구문분석 사이의 중간 과정이라 볼 수 있는 구문 요소 인식을 위한 일관된 모델을 제시할 수 있으며, 그에 대한 확장이 필요하다. 향후 연구 사항으로는 대용량의 자료에 대한 실험이 필요하며, 코퍼스의 실제 대역 내용을 정렬하여 일부 표현을 추출하여 이를 유한상태 변환기로 변환하는 방법에 대한 연구가 필요하다.

5 참고문헌

[1] 김정재, 박준식, 최기선. "두단계 대역어선택 방식을 이용한 구단위 패턴기반 한영 기계번역 시스템", 한글 및 한국어정보처리 학술대회, 1999 (발표예정)

[2] 백대호, 이호, 임해창. "Finite State Transducer를 이용한 한국어 전자 사전의 구조". 한글 및 한국어정보처리 학술대회, pp181-187, 1996.

[3] 안동언. "기계번역을 위한 한국어 해석에서 형태소로부터 구문요소의 형성에 관한 연구". 한국과학기술원, 석사학위논문, 1987.

[4] 은종진. "효율적인 구문분석을 위한 전처리기 구현과 복합 명사의 구조 분석". 한국과학기술원, 석사학위논문, 1996.

[5] 최기선 외. "한국어정보베이스를 위한 형태· 통사 태그 표준에 관한 연구". 인지과학, Vol. 7, No. 4, pp. 43-61, 1996.

[6] Alshawi, H. "Head Automata for Speech Translation". In *Proceedings of ICSLP-96*, 1996

[7] Alshawi, H. "Head Automata and tree tilling: translation with minimal representations". In *Proceedings of the 34th Anuual Meeting of the*

Association for Computational Linguistics. pp167-176. 1996

[8] Oncina, J., García, P., Vidal, E. "Learning Subsequential Transducers for Pattern Recognition Interpretation Tasks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 5. pp.448-458, May 1993

[9] Roche, E. and Schabes, S. *Finite-State Lanugauage Processing*, The MIT Press, 1997

[10] Vilar, J. M., Vidal, E. and Amengual, J. C. "Learning Extended Finite State Models for Language Translation". In *Proceedings of ECAI-96*, 1996

[11] Vilar, J. M., Castellanos, A., Jimemez, J. M., Sanchez, J. A., Vidal, E., Oncina, J., and Rulot, H. "Spoken-language machine translation in limited domains: can it be achieved by finite-state models ?". In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*. pp326-333. 1995