論文98-35C-10-5

통계 정보와 유전자 학습에 의한 최적의 문장 분할 위치 결정

(Determination of an Optimal Sentence Segmentation Position using Statistical Information and Genetic Learning)

金 聖 東 *, 金 榮 澤 *

(Sung Dong Kim and Yung Taek Kim)

요 약

실용적인 기계번역 시스템을 위한 구문 분석은 긴 문장의 분석을 허용하여야 하는데 긴 문장의 분석은 높은 분석의 복잡도 때문에 매우 어려운 문제이다. 본 논문에서는 긴 문장의 효율적인 분석을 위해 문장을 분할하는 방법을 제안하며 통계 정보와 유전자 학습에 의한 최적의 문장 분할 위치 결정 방법을 소개한다. 문장 분할 위치의 결정은 분할 위치가 태그된 훈련 데이타에서 얻어진 어휘 문맥 제한 조건을 이용하여 입력 문장의 분할 가능 위치를 결정하는 부분과 여러 개의 분할 가능 위치 중에서 안전한 분할을 보장하고 보다 많은 분석의 효율 향상을 얻을 수 있는 최적의 분할 위치를 학습을 통해 선택하는 부분으로 구성된다. 실험을 통해 제안된 문장 분할 위치 결정 방법이 안전한 분할을 수행하며 문장 분석의 효율을 향상시킴을 보인다.

Abstract

The syntactic analysis for the practical machine translation should be able to analyze a long sentence, but the long sentence analysis is a critical problem because of its high analysis complexity. In this paper a sentence segmentation method is proposed for an efficient analysis of a long sentence and the method of determining optimal sentence segmentation positions using statistical information and genetic learning is introduced. It consists of two modules: (1) decomposable position determination which uses lexical contextual constraints acquired from a training data tagged with segmentation positions. (2) segmentation position selection by the selection function of which the weights of parameters are determined through genetic learning, which selects safe segmentation positions with enhancing the analysis efficiency as much as possible. The safe segmentation by the proposed sentence segmentation method and the efficiency enhancement of the analysis are presented through experiments.

I. 서 론

기계번역을 위한 입력 문장 분석을 위하여 다음과 같은 여러 가지 방법이 연구되어 왔다: 규칙 기반(rule-based) 방법, 예문 기반(example-based) 방법[1,2], 통계적(statistical) 방법[3,4]. 또한 숙어를 이용하는 방법이 영한 기계번역에서 분석의 효율성과 번역의 정확성 향상을 위해 도입되었다[5,6]. 이 방법들은 주로 문장 분석에서 나타나는 모호성(ambiguity) 문제를 해결하는데 중점을 두고 있다.

실용적인 기계번역 시스템이 되기 위해서는 모호성 해결 이외에 긴 문장의 분석이 필수적이다. 문장이 길어질수록 분석의 시간/공간 복잡도가 증가하여 분석이 성공적으로 수행되기 어려우며, 분석이 종료한다 하더

* 正會員, 서울大學校 컴퓨터工學科
(Department of Computer Engineering, Seoul National University)

接受日字:1998年8月3日, 수정완료일:1998年9月25日

라도 정확한 번역을 얻기가 어렵다. 예문 기반 방법에서는, 문장이 길어질수록 원시 문장과 정확하게 일치하는 예문을 발견하기 어려워지기 때문에 만족할 만한 결과를 얻기 어렵다[7]. 또한 숙어를 이용하는 방법에서도, 문장이 길어질수록 숙어 인식의 복잡도가 증가하여 숙어 인식에서 많은 자원을 소모하게 되므로 긴 문장의 분석이 매우 어려워진다. 즉, 긴 문장이 비록 문법적인 오류를 가지지 않는다 하더라도 분석의 복잡도가 크기 때문에 정확한 분석 결과를 얻는 것은 매우 어렵다[8].

긴 문장을 보다 효율적으로 분석하여 정확한 결과를 얻기 위해 여러 가지 연구가 진행되어 왔다. 예문 기반 방법에서 도입된 구성 성분 경계 파싱(constituent-boundary parsing) 방법은 효율적인 분석을 하기 위해 패턴 매칭(pattern matching) 방법을 도입하였다[9]. 영어 문장 분석을 위해서 [10,11] 에서는 신경망(neural network)을 이용하여 문장을 [주어 앞 부분 + 주어 + 서술부] 로 분할하는 방법을 제안하였다. 이 방법은 주로 단문[1] (simple sentence)을 분석할 때 주어 앞 부분이나 주어가 긴 문장에 대해서 유용하다. [12] 에서는 긴 문장의 패턴을 정의하기 위하여 문장 패턴(sentence pattern)을 도입하였다. 문장은 정의된 패턴에 의해 분할되고 분할된 세그먼트들은 독립적으로 분석되어 문장 패턴이 명시한 규칙에 의해 결합되어 전체 문장에 대한 분석 결과가 생성된다. 이 방법은 정의한 문장 패턴에 맞는 문장에 대해서 보다 효율적인 분석을 수행한다. 불어-영어 기계번역에서는 영어 문장을 분할하기 위한 최대 엔트로피 모델(maximum entropy model)을 제안하였다[13]. 최대 엔트로피 모델이 분할 위치에 분할의 적절성(appropriateness)을 나타내는 점수(score)를 할당한다.

본 논문에서는 문맥 자유 문법(context free grammar)을 가지고 챠트 파싱(chart parsing)을 이용하는 영어-한국어 기계번역을 위한 구문 분석의 선처리 단계로서 긴 문장의 분석을 용이하게 하기 위한 문장 분할(sentence segmentation)을 제안하고 통계 정보(statistical information)와 유전자 학습(genetic learning)을 이용한 최적의 문장 분할 위치(optimal sentence segmentation position) 결정 방법을 소개한다. 분할 위치가 태그된 훈련 데이타로부터 얻어진 어휘 문맥 제한 조건으로 문장의 분할 가능 위치를 결정하고 유전자 학습을 통하여 여러 개의 분할 가능 위치 중에서 안전한 분할을 도모하고 보다 큰 분석 효율 향상을 얻을 수 있는 분할 위치를 선택한다. 일정한 길이 이상의 문장은 그것이 가진 분할 가능 위치 중에서 최적의 위치에서 분할되며 일정한 길이 이하의 세그먼트들이 생성될 때까지 분할이 계속된다. 문장 분할을 통해 효율적인 분석이 가능해 지지만 잘못된 문장 분할은 잘못된 분석 결과를 생성하므로 분할 위치의 선택은 안전한 분할을 고려해야 한다.

긴 문장은 쉼표(comma)를 포함한 문장과 그렇지 않은 문장으로 구분할 수 있으며 쉼표는 명시적인 문장 분할 위치가 될 수 있으므로 쉼표를 포함한 문장의 분할은 본 논문에서 고려되지 않는다. 즉 쉼표를 포함하지 않은 긴 문장의 분할이 본 논문에서 다루어지는 주요한 문제이다.

본 논문은 다음과 같이 구성되었다. 2장에서는 세그먼트를 정의하고 훈련 데이터로부터의 어휘 문맥 제한 조건 생성과 그것을 이용한 분할 가능 위치 결정 방법을 설명한다. 그리고 훈련 데이타에서 얻어진 통계 정보와 유전자 학습을 이용한 분할 위치 선택(segmentation position selection) 방법을 3장에서 설명한다. 4장 실험에서는 분할 위치 선택의 정확성을 보이며, 문장 분할을 이용한 분석의 시간/공간 면에서의 효율 향상을 제시하고 5장에서 본 논문을 결론짓는다.

II. 세그먼트와 분할 가능 위치

1. 세그먼트 (segment)

세그먼트는 문장의 구나 절에 대응하는 문장의 한 블럭(block)을 의미하며 문장의 구성 성분(constituent)이 된다. 즉 문맥 자유 문법[2](context-free grammar)을 구성하는 비단말 노드(nonterminal)에 대응된다. 세그먼트는 일련의 단어로 구성되거나 단어와 서브-세그먼트로 구성되며 k에서 l까지의 단어를

1) 하나의 주어와 하나의 서술부를 갖는 문장. 또는 기저 구조(underlying structure)에서 S(sentence) 하나만을 갖는 문장[14].

2) $<T, N, NI, R>$: $T = \{t^1, t^2, ..., t^w\}$, $N = \{N^1, N^2, ..., N^v\}$, N^1: 첫 기호, R: 규칙 집합 ($N^i \rightarrow \zeta$), $\zeta \in (T \cup N)$

지배하는 구문 범주 기호(syntactic category symbol)를 N^p 라고 할 때 세그먼트와 문장을 정의하면 (정의 1,2)와 같다.

(정의1) 세그먼트(segment)

$s^{N^p}_{k,l}=\{\ w_i,\ s^{N^p}_{m+1,l} \mid k\le i\le m,\ k<m\le l\ \},\ s^{N^p}_{k,k}=\ w_k,$

$N^P \rightarrow N^Q \ldots N^R,\ where\ N^P,\ N^Q,\ N^R \in N$

(정의 2) 문장(sentence)

$S = \{\ s^{N^p}_{k,l} \mid 1\le k\le n,\ 1\le l\le n,\ k\le l\ \}$

문장이 구와 절로 구성되므로 구와 절에 대응되는 복수개의 세그먼트가 문장을 구성한다. 그림 1은 문장 In_1 the_2 $morning_3$ the_4 boy_5 $walked_6$ in_7 the_8 $park_9$ 을 세그먼트 결합으로 표현한 문장 구조의 예를 보여준다.

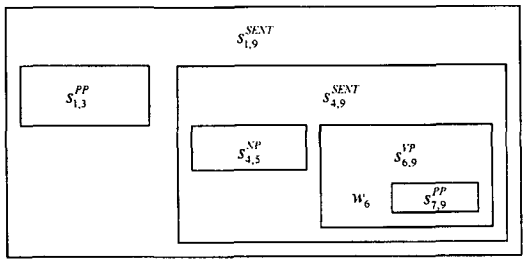

그림 1. 세그먼트 결합으로 표현된 문장 구조
Fig. 1. Sentence Structure represented by the Combination of Segments.

그림 1에서 문장 세그먼트($s^S_{1,9}$)는 전치사구 세그먼트($s^{PP}_{1,3}$)와 문장 세그먼트($s^S_{4,9}$)로 구성되며, 다시 문장 세그먼트($s^S_{4,9}$)는 명사구 세그먼트($s^{NP}_{4,5}$)와 동사구 세그먼트($s^{VP}_{6,9}$)로 구성된다. 또한 동사구 세그먼트($s^{VP}_{6,9}$)는 동사(w_6)와 전치사구 세그먼트($s^{PP}_{7,9}$)로 구성됨을 알 수 있다.

2. 분할 가능 위치 (decomposable positions)

분할 가능 위치는 세그먼트의 시작 위치 또는 세그먼트간의 경계가 될 수 있는 단어의 위치이다. 이것은 구성 성분의 경계 위치가 되며, 즉 특정한 구나 절의 시작 위치가 분할 가능 위치이다. 그림 1에서 the_4, $walked_6$, in_7의 위치가 분할 가능 위치이다. 분할 가능 위치를 결정하는 과정은 그림 2와 같다.

분할 가능 위치는 분할 위치가 태그된 훈련 데이타의 학습을 통해 얻어지는 어휘 문맥 제한 조건에 의하여 결정된다. 어휘 문맥 제한 조건을 생성하기 위해 그림 3과 같은 5개 단어 크기의 윈도우(window)를 포함하는 어휘 문맥(lexical context)을 정의한다: 왼쪽의 2개 단어, 현재 단어, 오른쪽의 2개 단어, 각각의 품사, 왼쪽 2개 단어의 하위 범주화(subcategorization) 정보, 현재 단어의 위치 비율[3].

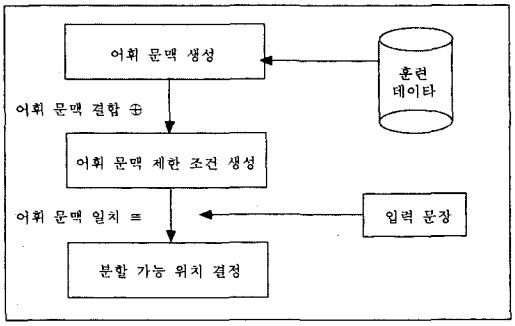

그림 2. 분할 가능 위치 결정 과정
Fig. 2. Steps of Decomposable Position Determination.

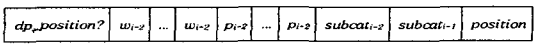

그림 3. 어휘 문맥
Fig. 3. Lexical Context.

표 1. 단어 집합과 예
Table 1. Word Set and Examples.

단어 집합	예
관계 대명사	*which, who, whose, ...*
의문사	*what, where, why, how, ...*
종속 접속사	*if, when, while, until, ...*
대등 접속사	*and, but, or, ...*
주격 대명사	*i, he, she, we, ...*
주격-목적격 대명사	*you, it*
조동사	*could, can, will, would, should, ...*
전치사1	*after, as, before, until, ...*
전치사2	*on, in, to, with, within, ...*
한정 형용사	*a, an, the, ...*

데이타 희소성(data sparseness)을 극복하기 위해

3) 위치 비율 = 현재 단어의 위치/문장의 단어 개수

표 1과 같이 단어 집합[4](word set)을 정의하여 단어 집합에 속하는 단어에 대해서는 그 집합에 대한 어휘 문맥을 구성한다. 단어 집합은 그 집합에 속하는 단어가 특정한 구성 성분의 첫 단어가 되거나 언어학적으로 유사한 특성을 가진다는 사실에 근거하여 정의된다.

단어에 대한 어휘 문맥은 활성(active) 문맥과 비활성(inactive) 문맥으로 구분되며 활성 문맥에 의해 분할 가능 위치를 결정하는 어휘 문맥 제한 조건이 결정된다. 분할 위치로 태그된 단어에 대해서는 활성 문맥 정보가 다른 단어에 대해서는 비활성 문맥 정보가 생성된다. 그림 4는 학습 데이터와 활성 문맥의 예를 보여준다.

어휘 문맥 제한 조건의 생성과 분할 가능 위치의 결정을 위해 어휘 문맥에 대한 결합(join)과 일치(consistency) 연산을 정의한다. '*'는 두 문맥에서 서로 다른 값을 가지는 특성 항목으로 비활성 특성(inactive feature)이며 어휘 문맥 일치 연산에서 무관항(don't care term)의 역할을 한다. 그리고 서로 같은 값을 가지는 항목은 활성 특성(active feature)이다.

You can change the panel names and the location of fields on the panels ***#if*** you leave the fields in the same order.

<1 RESTRIC_ADJ_SET panels ***SUB_CONJ_SET***[5] SUBJ_OBJ_PRON_SET leave 7708 5 7700 7705 3 0 0 0.62>

그림 4. 학습 데이타와 활성 문맥의 예[6]
Fig. 4. Example of Training Data and Active Context.

(정의 3) 어휘 문맥 결합 ⊕

$\langle x_1,...,x_n\rangle \oplus \langle y_1,...,y_n\rangle = \langle z_1,...,z_n\rangle$

z_i = * if $x_i \neq y_i$
x_i if $x_i = y_i$
$z_{min\_position} = min(x_{position}, y_{position})$

(정의 4) 어휘 문맥 일치 ≡

$\langle x_1,...,x_n\rangle \equiv \langle z_1,...,z_n\rangle$

(1) for all z_i ($1 \le i \le n$), z_i='*' or z_i=x_i

(2) $z_{min\_position} \le x_{position}$

$\langle x_1, ..., x_n\rangle$: 어휘 문맥
$\langle z_1, ..., z_n\rangle$: 어휘 문맥 제한 조건

어휘 문맥의 결합으로 생성되는 어휘 문맥 제한 조건은 비활성 특성과 활성 특성으로 이루어지며 그림 4의 항목 중 *position, dp_position* 항목을 제외한 12개의 항목으로 구성된다. 또한 어휘 문맥 제한 조건에는 결합되는 활성 문맥의 *position* 항목에 의해 결정되는 위치의 하한, 활성 특성의 개수, 그리고 분할 위치 선택시에 이용되는 어휘 문맥 제한 조건 확률(lexical contextual constraint probability)이 할당된다. 따라서 어휘 문맥 제한 조건은 15개의 항목으로 구성되며 처음 12개의 항목과 활성 특성의 개수는 문장의 분할 가능 위치 결정에 이용되고 활성 특성의 개수와 확률은 분할 위치 선택시 고려된다. 본 논문에서는 결합된 활성 문맥의 개수가 5개 이상인 어휘 문맥 제한 조건만을 생성하였다.

입력 문장의 분할 가능 위치 결정은 (정의 4)의 어휘 문맥 일치 연산에 의한다. 문장의 각 단어에 대해 어휘 문맥을 구성하고 단어의 어휘 문맥과 어휘 문맥 제한 조건과 일치를 시도한다. 한 단어의 어휘 문맥과 일치하는 어휘 문맥 제한 조건은 복수로 존재할 수 있으며 그 중 활성 특성의 개수가 가장 많은 어휘 문맥 제한 조건이 선택된다. 어휘 문맥 일치에 성공한 단어의 위치는 분할 가능 위치가 되며 선택된 어휘 문맥 제한 조건으로 표현된다.

(정의 5) 어휘 문맥 제한 조건 확률 (Lexical Contextual Constraint Probability)

$$p(lcc_i) = \frac{Count(active\ context_1)}{Count(active\ context_2) + Count(inactive\ context_3)}$$

Count() : 개수를 세는 함수
active context$_1$: lcc_i 를 생성하는데 결합(⊕)된 활성 문맥
active context$_2$: 단어에 대한 모든 활성 문맥
inactive context$_3$: lcc_i 와 일치(≡)하는 비활성 문맥

Ⅲ. 분할 위치 선택(segmentation position selection)

1. 안전한 분할 (safe segmentation)

문장 분할의 목적은 긴 문장의 분석을 용이하게 하여 보다 효율적이고 정확한 분석을 하는 것이다. 문장 분할이 긴 문장을 짧은 복수개의 세그먼트로 변환하여 문장 분석이 하나의 긴 문장에 대해 이루어지지 않고 세그먼트들에 대해 이루어지므로 분석의 복잡도가 줄

4) 전치사1 : 종속 접속사로도 사용될 수 있는 전치사
한정 형용사 : 제한적 용법으로만 사용되는 형용사와 관사를 포함한다

6) #는 분할 위치를 의미한다

고 효율을 향상시킬 수 있다는 것은 직관적으로 알 수 있는 사실이다. 그러나 잘못된 문장 분할로 인해 잘못된 분석 결과가 생성될 수 있으므로 복수개의 분할 가능 위치 중에서 안전한 분할을 고려한 분할 위치 선택이 매우 중요하다.

관련성 있는 단어의 블럭을 생성하는 분할을 안전한 분할이라 정의할 수 있다. 그러나 이 정의는 기계번역의 대상이 되는 원시 언어(source language)와 목적언어(target language)의 관계에 따라 다르게 정의되어야 한다. 영어-한국어 기계번역을 위해서 본 논문에서 정의된 세그먼트는 문장의 구나 절에 대응되는 개념이고 분석에서 생성된 구조는 트리(tree) 모양의 계층적 구조를 이루게되며 따라서 안전한 분할은 이러한 계층 구조를 유지할 수 있도록 세그먼트를 생성해야 한다. 즉, 안전한 분할은 안전한 세그먼트를 생성하는 분할로 정의할 수 있으며 안전한 세그먼트는 (정의 6)과 같다. 세그먼트 $s_{k,l}$를 직접 또는 간접적으로 지배하는 구문 범주 기호 N^P가 존재하고 앞의 세그먼트와 결합되어 분석될 수 있어야 한다. 본 논문에서는 안전한 세그먼트를 생성하는 분할을 안전한 분할이라 정의한다.

(정의 6) 안전한 세그먼트 (safe segment)
segment $s_{k,l}$ *is safe iff*

(1) $N^P \Rightarrow w_k \ldots w_l$, $N^P \in N$, $k \le l$
(2) $N^Q \Rightarrow s_{i,j}\ s_{k,l}$, $N^Q \in N$, $j = k-1$, $1 \le i \le j$

=>는 직접 또는 간접 지배 (direct or indirect dominance) 관계

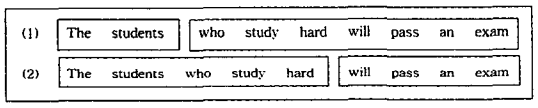

그림 5. 불안전한 분할과 안전할 분할의 예
Fig. 5. Example of Unsafe/Safe Segmentation.

그림 5는 문장 *The students who study hard will pass an exam*에 대한 안전한 분할과 불안전한 분할을 보여준다. 그림 5의 (1)은 불안전한 분할이고 (2)는 안전한 분할이다. 분할 (1)의 결과로 생성되는 두개의 세그먼트를 분석하였을 때 두번째 세그먼트는 관계절(RLCL)과 동사구(VP)로 분석되는데 RLCL과 VP를 직접 지배 하는 규칙이 없기 때문에 정확한 분석을 얻을 수 없다. 반면에 분할 (2)의 결과로 생성되는 세그먼트를 분석하였을 때, 첫번째 세그먼트는 명사구(NP)를 생성하고 두번째 세그먼트는 동사구(VP)를 생성하므로 올바른 분석을 얻을 수 있다.

2. 분할 위치 선택

긴 문장의 분석을 용이하게 하기 위한 문장 분할은 분석의 효율을 향상시키는 동시에 안전한 분할을 고려해야 한다. 이를 위해 효율성 척도와 안전성 척도를 고려한 분할 적절성 함수(segmentation appropriateness function)를 정의하고 이것에 의하여 계산되는 점수에 따라 분할 위치를 선택한다. $O(n^3)$으로 표현되는 길이가 n인 문장의 분석 복잡도는 n에 의존하므로 길이 n을 줄이는 것이 효율성 향상에 주요한 요소이다. 효율성 척도는 분할된 세그먼트의 크기를 이용하여 계산되며 안전성 척도는 각 분할 가능 위치를 표현하는 어휘 문맥 제한 조건에 의하여 계산된다. 분할 적절성 함수는 효율성 척도와 안전성 척도의 선형 결합으로 표현되며 이 두 가지 척도를 고려한 분할 가능 위치의 분할 적절성 점수를 계산한다. 분할의 적절성은 보다 많은 분석 효율 향상을 위해 분할의 결과로 생성되는 세그먼트의 최대 크기가 작은 것을 선호하며 균등한 크기의 세그먼트 생성을 선호한다. 그리고 안전한 분할을 위해서 가능한 한 문장의 뒤에서 분할하는 것이 좋다. 또한 훈련 데이터에서 다른 단어에 비해 더 자주 분할 위치로 선택된 단어를 선호하며 분할 가능 위치를 표현하는 어휘 문맥 제한 조건의 확률과 명확성(specification)을 고려한다. 따라서 분할 적절성 함수는 분할된 세그먼트의 최대 크기, 분할된 세그먼트의 크기 차, 분할 위치 단어의 분할 확률[7], 활성 특성의 개수, 분할 가능 위치의 문장에서의 위치 비율, 그리고 분할 가능 위치의 어휘 문맥 제한 조건 확률 등 6개의 인자를 가진 선형 함수로 표현되며 유전자 학습(genetic learning) 알고리즘을 이용하여 각 변수의 가중치(weight)를 결정하였다. 분할 적절성 함수의 각 인자가 동등하게 분할 위치 결정에 참여한다는 가정 하에 인자의 값은 모두 0에서 1까지의 값을 갖도록 정규화 하였다. 유전자 학습을 위한 염색체(chromosome)는 6개의 가중치를 원소로 하는 벡터가 된다. 적합도(fitness) 함수는 (정의 7)과 같으며 최대 세대수(maximum generation)를 5000으로 제

7) 단어의 활성 문맥의 개수/(단어의 활성 문맥의 개수 + 단어의 비활성 문맥의 개수)

한하고 모집단의 최대 적합도의 값이 50 세대동안 지속되는 지의 여부를 학습의 종료 조건으로 한다. 초기 모집단을 임의로 생성하고 각 염색체의 적합도를 계산하고 종료 조건을 만족시킬 때까지 다음의 과정을 반복한다.

(1) (정의 8)의 선택 확률에 따라 모집단에서 일정한 개수의 염색체를 선택하여 다음 세대를 생성한다.
(2) 모집단에서 선택 확률에 따라 일정한 개수의 염색체쌍을 선택하여 이들의 교차(crossover) 연산에 의해 자식(offspring)을 생성하여 다음 세대에 추가한다.
(3) 다음 세대에서 일정한 부분을 선택하여 돌연변이(mutation) 연산을 적용시킨다.
(4) 새로 생성된 다음 세대로 모집단을 갱신한다.

(정의 7) 적합도 함수

$$fitness(h^{g)}) = \frac{\#\ of\ satisfied\ training\ data\ by\ h}{\#\ of\ training\ data}$$

(정의 8) 선택 확률

$$pr(h_i) = \frac{fitness(h_i)}{\sum_{j=1}^{n} fitness(h_j)}$$

위의 실험을 100회 반복하여 가장 적합도가 높은 개체의 가중치를 선택하였다. 유전자 학습 알고리즘을 위한 주요 인자 값은 표 2와 같다.

표 2. 유전자 알고리즘을 위한 인자값
Table 2. Parameter Values used in Genetic Algorithm.

인자	값
모집단 크기(population size : p)	100
교차율 (crossover rate : r)	0.9
돌연변이율 (mutation rate : m)	0.1
최대 세대수(maximum generation)	5000

분할 적절성 함수는 문장의 모든 분할 가능 위치(decomposable position : dp로 표현)의 분할 적절성 점수를 계산하고 이 중 가장 큰 점수를 가진 위치(dp^*)가 분할 위치로 선택된다. 분할 적절성 함수와 분할 위치 선택은 아래와 같이 정의된다.

(정의 9) 분할 적절성 함수(segmentation appropriateness function)

$$SAF(dp^i) = \sum_{k=1}^{m} w_k^i v_k^i$$

(정의 10) 분할 위치 선택(segmentation position selection)

$$dp^* = \underset{dp^i \in D}{\arg\max}\ SAF(dp^i)$$

Ⅳ. 실험

1. 말뭉치 (corpus)

말뭉치는 어휘 문맥 제한 조건의 획득을 위한 데이타, 유전자 학습 알고리즘을 위한 데이타, 그리고 분할 모델의 성능 평가를 위한 테스트 데이타로 구분된다.

어휘 문맥 제한 조건의 획득을 위해 쉼표(comma)를 포함하지 않는 길이 15 이상의 고등학교 교과서에서 4480개, 컴퓨터 매뉴얼에서 1500개의 문장을 추출하여 사람이 분할 위치를 태그하여 훈련 데이타를 구축하였으며 그것으로부터 265개의 어휘 문맥 제한 조건을 생성하였다. 분할 적정성 점수 계산에 포함되는 가중치의 조정을 위한 유전자 학습을 위해 컴퓨터 매뉴얼과 고등학교 영어 교과서에 나타나는 길이 15~35의 500 문장을 이용하였다. 그리고 문장 분할의 성능 평가를 위해 표 3과 같은 길이 분포를 가지는 훈련 데이타와 같은 영역, 컴퓨터 영역의 바이트 매거진(Byte magazine), 그리고 워싱턴 포스트지(Washington Post)에서 쉼표를 포함하지 않는 각각 600문장을 추출하였다. 그리고 문장 분할을 이용한 분석의 시간/공간 면에서의 효율 향상을 측정하기 위해 같은 문장 집합을 이용하였다.

표 3. 테스트 문장의 영역과 길이 분포
Table 3. Domain and Length Distribution of the Test Sentences.

	문장 길이	문장 수
교과서 + 컴퓨터 매뉴얼	15 ~ 19	200
	20 ~ 24	200
	25 ~ 29	150
	30 ~	50
바이트 매거진	15 ~ 19	200
	20 ~ 24	200
	25 ~ 29	150
	30 ~	50
워싱턴 포스트	15 ~ 19	200
	20 ~ 24	200
	25 ~ 29	150
	30 ~	50

2. 분할 성능 (segmentation performance)

분할 성능은 분할의 적용률(recall)과 정확도(precision)로 표현된다. 적용률은 일정한 길이[9] 이상의 문장으로서 분할의 대상이 되는 문장 수와 실제로

분할된 문장 수의 비로 표현되며 정확도는 실제로 분할된 문장 중 안전하게 분할된 문장 수의 비로 표현된다. 적절한 크기의 세그먼트로 분할되지 않은 문장과 불안전한 분할에 의해 분할된 문장을 분할 오류 문장(sentences with segmentation error)이라 정의하고 분할 오류가 발생하지 않은 문장의 수와 전체 문장의 수를 이용하여 분할 성능을 계산한다.

(정의 11) 적용률 (recall)

$$recall = \frac{\#\ of\ actually\ segmented\ sentence}{\#\ of\ sentence\ to\ be\ segmented}$$

(정의 12) 정확도 (precision)

$$precision = \frac{\#\ of\ sentences\ with\ safe\ segmentation}{\#\ of\ actually\ segmented\ sentences}$$

(정의 13) 분할 성능 SP

$$SP = \frac{\#\ of\ sentences\ to\ be\ segmented - \#\ of\ sentences\ with\ segmentation\ error}{\#\ of\ sentences\ to\ be\ segmented}$$

각 영역에서의 적용률, 정확도 그리고 분할 성능을 문장 길이별로 구분하면 표 4와 같다. 훈련 데이타와 같은 영역에서 추출된 테스트 문장에 대해서는 약 93 %의 분할 성능을 보이고 바이트 매거진의 문장에 대해서는 약 90 %, 그리고 워싱턴 포스트지의 문장에 대해서는 88 %의 분할 성능을 보인다. 영역에 따라 분할 성능에 차이가 있지만 전체적으로 약 90 %의 분할 성능은 영역과 무관하게 본 논문에 제시된 최적의 분할 위치 결정 방법이 긴 문장의 효율적인 분석을 위해 적용될 수 있다는 것을 의미한다.

3. 분할 오류 분석

분석 오류는 3가지의 부류로 나눌 수 있다. 첫째는 불안전한 분할이다. 예를 들면 문장, *The developers created the objects as 3-D models and then rendered them into bit maps from various angles*은 [*The developers created the objects*], [*as 3-D models and then rendered them into bit maps from various angles*] 의 2개의 세그먼트로 분할되는데 두번째 세그먼트는 하나의 구나 절로 분석되지 않기 때문에 전체 문장 분석이 실패하게 된다. 이는 주로 선택 함수에 의한 분할 위치의 선택이 잘못된 경우이다. 둘째, 일정한 크기 이상의 세그먼트인데 분할되지 않은 경우이다. 예를 들어, *These two objects are the result of the cooperation of the virtual-memory manager and an external pager* 는 크기가 17임에도 불구하고 분할되지 않았다. 이는 주로 훈련 데이터를 통해 얻어진 어휘 문맥 제한 조건과 일치하는 분할 가능 위치를 발견하지 못하였기 때문이다. 세째, 선택된 분할 위치가 비록 안전한 분할일지라도 더 좋은 분할 위치[11]가 존재하는 경우이다. 예를 들면, *Fully 10 percent of those polled indicated they had made a final decision as to their presidential choice*는 [*Fully 10 percent of those polled indicated they had made a final decision*], [*as to their presidential choice*] 로 분할되고 첫번째 세그먼트는 다시 [*Fully 10 percent of those polled indicated*], [*they had made a final decision*] 으로 분할된다. 첫번째 분할 위치로 *they*가 선택되었으면 한번의 분할으로 적절한 크기의 세그먼트가 생성된다. 2절에서 계산된 분할 오류 문장의 수에는 첫번째와 두번째 유형의 오류 문장의 수만이 포함되었다. 첫번째 유형의 오류는 잘못된 분석을 유발하고 두번째 유형의 오류는 긴 세그먼트를 분할하지 않음으로써 분석이 종료하지 못할 수 있기 때문에 본 논문에서 제안된 문장 분할의 목적에 부합하지 않는 결과를 초래하기 때문이다.

표 4. 문장 길이에 따른 분할 성능[10]
Table 4. Segmentation Performance classified by the Sentence Length.

	문장 길이	오류1	오류2	적용률	정확도	분할 성능
교과서 + 컴퓨터 매뉴얼	15 ~ 19	2	10	99 %	94.9 %	94 %
	20 ~ 24	3	11	98.5 %	94.4 %	93 %
	25 ~ 29	3	4	98 %	97.3 %	95.3 %
	30 ~	5	3	90 %	93.3 %	84 %
바이트 매거진	15 ~ 19	4	16	98 %	91.8 %	90 %
	20 ~ 24	6	15	97 %	92.3 %	89.5 %
	25 ~ 29	4	5	97.3 %	89.1 %	94 %
	30 ~	5	4	90 %	91.1 %	82 %
워싱턴 포스트	15 ~ 19	7	15	96.5 %	92.2 %	89 %
	20 ~ 24	3	22	98.5 %	88.8 %	87.5 %
	25 ~ 29	2	11	98.7 %	92.6 %	91.3 %
	30 ~	5	8	90 %	82.2 %	74 %
전체	1800 문장	49	124	97.3 %	92.9 %	90.4 %

9) 적절한 세그먼트의 크기는 11로 정하였다

11) 보다 세그먼트를 균등하게 분할하는 위치 또는 전체 분할 횟수를 줄일 수 있는 분할 위치

4. 시간/공간 면에서의 분석 효율 향상

문장 분할에 의한 분석의 효율 향상을 분석에 소요된 시간과 공간면에서 측정하였다. 분할을 하지 않고 분석하는 경우에는 문장 길이가 20이 넘으면 많은 경우에 분석이 종료하지 못하기 때문에 15에서 19 사이의 문장에 대해서만 효율 향상을 측정하였다. 문장 분석은 Sun-Sparc 20 시스템에서 수행되었고 분할에 의한 분석과 그렇지 않은 분석에 의한 실험 결과는 표 5와 같으며 효율 향상은 (정의 14)에서 정의된다.

표 5에 나타난 수치는 한 문장을 분석할 때 소요되는 시간과 공간의 평균값을 의미한다. 분할을 이용하지 않았을 때 교과서와 컴퓨터 영역의 170 문장, 바이트 매거진의 158 문장, 그리고 워싱턴 포스트의 136 문장에 대해서만 분석이 종료되었다. 그러므로 분할을 이용한 분석과의 비교는 분석이 종료한 문장에 대하여 이루어졌다.

(정의 14) 효율 향상 (PI : performance improvement)

$$PI_{time} = \frac{t_{unseg} - t_{seg}}{t_{unseg}} \times 100 \qquad PI_{memory} = \frac{m_{unseg} - m_{seg}}{m_{unseg}} \times 100$$

t_{unseg}, m_{unseg} : *used time/memory during analysis without segmentation*
t_{seg}, m_{seg} : *usedtime/memory during analysis with segmentation*

표 5. 분할에 의한 분석과 분할을 이용하지 않은 분석의 결과

Table 5. Comparison of Analysis Results with/ without Segmentation.

		분할을 이용한 분석	분할을 이용하지 않은 분석	효율 향상
교과서 + 컴퓨터 매뉴얼	시간	6.5 sec	24.5 sec	73.5 %
	공간	1.3 MB	4.2 MB	69 %
바이트 매거진	시간	7.7 sec	31.3 sec	75.3 %
	공간	1.6 MB	4.6 MB	53.3 %
워싱턴 포스트	시간	7.3 sec	36.2 sec	80 %
	공간	1.5 MB	5.4 MB	72.2 %

5. 기존의 방법과의 비교

본 논문에서 제안된 최적의 분할 위치 결정 방법은 영한(英韓) 기계번역에서 긴 문장의 효율적인 분석을 목적으로 한다.

영일(英日) 기계번역에서 같은 목적을 위하여 제안된 구성 성분 경계 파싱[9]에서는 문법 정보를 패턴에 결합시킴으로써 분석 규칙의 개수를 줄이는 효과를 얻을 수 있고 따라서 분석에 이용되는 규칙의 개수가 적어져 보다 효율적인 분석이 가능하다. 여기서는 약 350개의 패턴을 도입하여 주로 10 단어 이내의 짧은 문장의 분석을 수행하였다. 따라서 긴 문장에 대해서도 효율적인 분석이 가능한 지는 미지수이다.

[10,11]에서는 신경망의 패턴 매칭 능력을 이용하여 문장을 [주어 앞 부분 + 주어 + 서술부]로 분할한 후 세 부분을 독립적으로 분석한다. 비교적 적은 훈련 데이타(300문장)를 가지고 주어의 중심어와 경계를 찾는데 약 97%의 성공률을 보인다. 하나의 문장을 세 부분으로 분할하여 분석을 하므로 효율적인 분석을 할 수 있으나 단문의 경우에만 적용할 수 있다. 긴 문장은 주로 2개 이상의 주어를 가진 대등 접속문과 복합문이며 이 방법은 2개 이상의 주어를 찾는데는 적용되지 않기 때문에 긴 문장 분석을 위한 방법으로는 적절하지 않다.

[12]에서는 긴 문장의 구성 형식을 문장 패턴으로 정의하여 문장 패턴에 의한 문장 분할과 분석을 수행한다. 이 방법을 통해 문장 분할을 이용하지 않는 분석 방법에 비해 시간 면에서 30.9%, 공간 면에서 57.8%의 효율 향상을 얻을 수 있다. 이는 본 논문에서 제시된 결과와 비교하여 시간 면에서 약 40%, 공간 면에서 약 10% 떨어짐을 알 수 있다. 또한 문장 패턴의 적용률(recall)이 36.2%에 불과하여 긴 문장 분석을 위한 일반적인 방법으로는 적절하지 않음을 알 수 있다.

[13]에서는 불영(佛英) 기계번역에서 번역 속도 향상을 위한 문장 분할을 위해 최대 엔트로피 모델을 제안하였다. 불영 기계번역에서는 영한 기계번역과는 달리 입력 문장의 분석이 필요하지 않으며 단지 적절한 대역어를 선정하여 번역을 수행한다. 따라서 문장 분할의 결과로 생성되는 세그먼트들은 논리적으로 관련있는 단어의 집합이 아니고 단지 왼쪽에서 오른쪽으로 순차적으로 번역될 수 있는 세그먼트이다. 그러나 영한 기계번역을 위해서는 생성된 세그먼트들이 논리적으로 관련있는 단어의 집합으로서 문장 구성 성분의 역할을 할 수 있어야 한다.

본 논문에 제안된 분할 위치 결정 방법은 문장의 종류나 길이에 무관하게 약 97%의 적용률을 보이므로 실용적인 영한 기계번역을 위해 사용될 수 있다. 문장 길이가 20 이상인 문장들을 분할을 이용하여 분석할 수 있었으며 문장 길이에 대한 분석 시간과 소요 메모리는 표 6과 같다. 문장 길이가 20이 넘으면 분할을

이용하지 않고서는 분석의 복잡도로 인해 분석을 종료하지 못하는 경우가 대부분인데 분할을 이용하면 분석을 종료할 수 있다. 표 6을 보면 문장이 길어질수록 분할을 이용하더라도 분석에 상당히 많은 시간이 소요됨을 알 수 있는데, 이를 해결하기 위해서는 분할 위치의 특성과 생성된 세그먼트간의 병렬성을 이용하는 효율적인 분석 알고리즘에 대한 연구가 필요하다.

표 6. 길이 20 이상의 문장에 대한 분할을 이용한 분석의 소요 시간과 메모리
Table 6. Time/Memory of Analysis using Segmentation for Sentences with more than 20 Words.

	길이 분포	평균 시간	평균 메모리
교과서 + 컴퓨터 매뉴얼	20 ~ 24	11.2 sec	2.7 MB
	25 ~ 29	19.6 sec	4.1 MB
	30 ~	40.1 sec	9.3 MB
바이트 매거진	20 ~ 24	14.6 sec	2.9 MB
	25 ~ 29	22.3 sec	4.3 MB
	30 ~	43.2 sec	10. 2 MB
워싱턴 포스트	20 ~ 24	15.1 sec	3.0 MB
	25 ~ 29	24.3 sec	4.7 MB
	30 ~	42.1 sec	9.8 MB

V. 결 론

본 논문에서는 긴 문장의 효율적인 분석을 위하여 문장 분할을 제안하고 통계 정보와 유전자 학습을 이용한 최적의 분할 위치를 결정하는 방법을 설명하였다. 4장에서 제시된 분할 성능과 분석 효율의 향상은 제시된 분할 위치 결정 방법이 실용적인 기계번역 시스템에 적용될 수 있다는 것을 보여주며, 긴 문장의 분석을 위한 방향을 제시한다는데 의의가 있다고 할 수 있다.

최적의 분할 위치를 결정하기 위하여 분할 위치가 태그된 훈련 데이터로부터 분할 가능 위치를 규정하는 어휘 문맥 제한 조건을 획득하였으며 유전자 학습 알고리즘을 통해 최적의 분할 위치를 선택하기 위한 분할 적절성 함수의 가중치를 조정하였다. 어휘 문맥 제한 조건은 사람이 결정하는 분할 위치의 특성을 표현한 것이라 할 수 있으며 유전자 학습 알고리즘은 여러 개의 분할 가능 위치가 존재할 때 사람이 분할 위치를 선택하는 기준의 학습을 통하여 신뢰성 있는 가중치를 생성한다.

훈련 데이터의 부족으로 인해 분할 가능 위치를 발견하지 못하는 경우에는 문장이 분할되지 않아서 분석을 종료하지 못하게 되고 분할 적절성 점수가 잘못 할당되어서 적절한 분할 위치가 선택되지 않는 경우에는 잘못된 분석 결과를 생성하는 결과를 초래할 수 있다. 문장에서 분할 가능 위치를 발견하지 못하는 경우, 어휘 문맥 제한 조건과의 일치 연산을 수정한 새로운 연산을 통해 분할 가능 위치를 발견하는 것이 필요하다. 이 새로운 연산은 일치 조건을 완화한 연산일 수 있다. 또는 분할 가능 위치를 결정하는 보조 규칙을 이용하는 것도 한 방법이 될 수 있다. 그리고 불안전한 분할을 분석 이전에 인식하여 안전한 분할을 얻을 수 있는 방법에 대한 연구가 필요하다. 동시에 보다 효율적이고 정확한 분석을 위해서 각 분할 위치의 특성에 따른 분석 알고리즘의 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] Nagao M., *A framework of a mechanical translation between Japanese and English by analogy principle.* A. Elithorn and R. Banerji (eds.), North-Holland, Amsterdam, 1984.

[2] Taijiro Tsutsumi and Hideo Watanabe *et al.*, "Example-based approach to Machine Translation", In *Proceedings of Japan-France MT workshop*, March 15-16, 1993.

[3] Brown P.F. et al., "A Statistical Approach to Machine Translation", *Computational Linguistics*, vol. 16, no. 2, pp. 79-85, 1980.

[4] P.Brown, J.Cocke, and S.Della Pietra *et al.*, "A Statistical Approach to Language Translation", In *Proceedings of the 1988 COLING*, pp. 71-76, Aug. 22-27, 1988.

[5] Sung Hee Yoon, "Efficient Parser to Find Bilingual Idiomatic Expressions for English-Korean Machine Translation." In *Proceedings of the 1994 ICCPOL*, pp. 455-460, May 10-13, 1994.

[6] Ho Suk Lee, "Automatic Construction of Transfer Dictionary based on the Corpus

for English-Korean Machine Translation", PhD thesis, Seoul National University, 1993. In Korean.

[7] Lambros CRANIAS, Harris PAPAGEORGIOU, and Stelios PIPERIDIS, "A Matching Technique in Example-Based Machine Translation", In *Proceedings of 1994 COLING*, pp. 100-104, 1994.

[8] Tetsura Nasukawa, "Robust Parsing Based on Discourse Information", In *33rd Annual Meeting of the ACL*, pages 33-46, 1995.

[9] Osamu FURUSE and Hitoshi IIDA, "Constituent Boundary Parsing for Example-Based Machine Translation", In *Proceedings of 1994 COLING*, pp. 105-111, 1994.

[10] Caroline Lyon and Bob Dickerson, "Reducing the complexity of parsing by a method of decomposition", In *International Workshop on Parsing Technology*, September, 1997.

[11] Caroline Lyon and Ray Frank, "Neural network design for a natural language parser", In *International Conference on Artificial Neural Networks*, 1995.

[12] Sung Dong Kim and Yung Taek Kim, "Sentence Analysis using Pattern Matching in English-Korean Machine Translation", In *Proceedings of the 1995 ICCPOL*, Oct. 25-28, 1995.

[13] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Pietra, "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, vol. 22, no. 1, pp. 39-72, 1996

[14] 이정민, 배영남, 언어학 사전. 박영사, 1990

저 자 소 개

金 聖 東(正會員)

1991년 서울대학교 컴퓨터공학과 학사. 1993년 서울대학교 컴퓨터공학과 석사. 1993년 ~ 현재 서울대학교 컴퓨터공학과 박사과정 재학중. 관심분야는 자연언어처리(Natural Language Processing), 기계번역(Machine Translation)

金 榮 澤(正會員)

1963년 미국 Colorado대 전기과 석사. 1968년 미국 Utah대 전산과 공학박사. 1975년 서울대학교 전자계산소 설치(소장 역임), 알골 컴파일러 완성. 1979년 ~ 1981년 미국 Purdue대, Yale대, Ilinois대 객원 교수. 1981년 한국 정보과학회 회장. 1990년 한국 인지과학회 회장. 현재 서울대학교 컴퓨터공학과 교수로 재직중. 관심분야는 프로그래밍 언어(Programming Language), 컴파일러(Compiler), 자연언어처리(Natural Language Processing)