

단어간 의존관계에 기반한 언어모델링*

이승미, 최기선

한국과학기술원 전산학과, 인공지능연구센터

대전시 유성구 구성동 373-1

우:305-701,

{leesm, kschoi}@world.kaist.ac.kr

Language Modeling based on Inter-Word Dependency Relation

Seungmi Lee, Key-Sun Choi

Department of Computer Science, Center for Artificial Intelligence

Korea Advanced Institute of Science and Technology

요약

확률적 언어모델링은 일련의 단어열에 문장확률값을 적용하는 기법으로서 음성 인식, 확률적 기계번역 등의 많은 자연언어처리 응용시스템의 중요한 한 요소이다. 기존의 접근방식으로는 크게 n-gram 기반, 문법 기반의 두가지가 있다. 일반적으로 n-gram 방식은 원거리 의존관계를 잘 표현할 수 없으며 문법기반 방식은 광범위한 커버리지의 문법을 습득하는데에 어려움을 가지고 있다. 본 논문에서는 일종의 단순한 의존문법을 기반으로 하는 언어모델링 기법을 제시한다. 의존문법은 단어와 단어 사이의 지배-피지배 관계로 구성되며 본 논문에서 소개되는 의존문법 재추정 알고리즘을 이용하여 원시 코퍼스로부터 자동적으로 학습된다. 실험 결과, 제시된 의존관계기반 모델이 tri-gram, bi-gram 모델보다 실험코퍼스에 대해서 약 11%에서 11.5%의 엔트로피 감소를 보임으로써 성능의 개선이 있었다.

1 서론

*본 연구는 정보통신부의 지원을 받아 수행중인 [지능형 멀티미디어 통합 정보베이스] 과제의 일환으로 이루어졌다.

언어 모델링은 주어진 단어열에 대해 선험적(*a priori*) 확률값을 지정하는 것으로 음성인식, 기계번역과 같은 응용에서 시스템의 성능을 결정하는 중요한 모듈이다. 언어 모델링을 위한 기존의 방법은 크게 두가지 부류로 나눌 수 있다. 일련의 $n-1$ 개의 단어열로부터 다음 n 번째 단어의 발생확률을 추정하는 n-gram 방식과 구문 문법을 이용하는 문법기반 언어 모델링이다.

N-gram 모델은 단어열 확률을 그 단어열 안의 각 단어의 확률의 곱으로 추정한다. 또, 각 n 번째 단어의 확률이 바로 전 $n-1$ 단어열에 의존한다고 가정한다. 이러한 n-gram은 주어진 학습코퍼스로부터 단순 빈도수 계산으로 학습될 수 있기 때문에 실용적으로 널리 이용되고 있다. 때로는 단어 대신 단어의 분류(tag, class) n-gram을 이용하는 경우도 있다[1, 2]. 그런데, 이 n-gram 모델은 그 가정에 의해, 원거리 의존관계를 표현할 수 없기 때문에 언어의 특성에 잘 부합되지 않는다.

이에 반하여, 최근에는 n-gram보다 언어학적 지식인 문법의 학습을 통한 문법기반 언어모델링이 시도되고 있다. 이는 통칭 구조화된 언어모델링이라 한다. 즉, 일련의 단어열에 구문구조를 할당하고, 그 구조를 이용하여 확률값을 구하는 방법이다. 원거리 의존관계는 이러한 구문구조를 통해 잘 표현될 수 있다. 사용되는 문법은 문맥자유문법, 재귀적천이망 등[5, 10, 3]의 구구조문법이다. 그런데 이 경우 해당 문법에 의해 문장으로 받아들여지지

않은 문장의 경우에는 확률값이 0이 된다. 따라서 어떤 문장도 0의 확률값이 되지 않도록 하려면, 문법이 광범위한 문형과 단어를 처리할 수 있을 만큼 되어야 하는데 이러한 문법의 확보가 매우 어렵다. 문법획득의 어려움 때문에, 문법으로 처리되지 않는 경우에는 n-gram 방식을 쓰거나, 혹은 두가지가 서로 보완하도록 결합된 모델을 구축하는 시도도 있다[7, 8]. 그럼에도, 아직까지 보편적으로 쓰이는 것은 바이그램이나 트라이그램이다. 물론 적절한 문법이 확보된다면, 이를 이용한 언어모델링은 기존의 n-gram 모델보다 훨씬 간결한 크기의 모델, 그리고 더 효과적인 모델이 될 것으로 예상된다.

본 논문에서는 n-gram의 학습 용이함과 강건함을 가지면서 언어적인 특성인 원거리 의존관계를 잘 표현할 수 있는 방법으로 단어간 의존관계에 의한 언어모델링을 제시한다. 이 단어간 의존관계는 단순한 형태의 의존문법으로부터 기인하며, 원형 코퍼스로부터 의존문법을 학습하는 확률의존문법 재추정알고리즘을 이용하여 자동으로 학습된다. 학습된 의존관계에 의해 문장이 분석되고 구문구조가 할당되어 이를 통해 원거리 의존관계가 잘 표현될 수 있다. 문법 획득이 자동화되고, 따라서 도메인 적응성을 높임으로써 결과적으로 실용성과 정확성이 서로 보완된 새로운 모델을 구축할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서 의존문법과 학습 알고리즘의 전개를 위한 정의와 표기법에 대해 설명하고, 3절에서 확률의존문법 학습 알고리즘을 설명한다. 4절에서는 제시된 모델과 n-gram 모델과의 비교실험 결과를 설명하고 5절에서 결론을 맺는다.

2 단어간 의존관계

본 논문에서는 아주 단순한 형태의 의존문법을 가정한다. 이 의존문법은 임의의 단어와 단어 사이의 지배-피지배 관계로 한 언어를 기술한다. 문장 분석은 지배-피지배 관계에 있는 단어들 사이의 의존링크를 연결함으로써 이루어진다. 한 문장의 의존관계분석, \mathcal{D} 는 그림 1에서와 같이 지배 단어에서 피지배 단어로의 화살표로 표현될 수 있다. 구조적 일반화를 위해 문장 끝에 항상 EOS(End Of Sentence)라는 표지가 있다고 가정하고, 또 문장 전체의 지배소가 EOS의 유일한 피지배소가 된다고 가정한다. 그림 1에서 단어 *gave*는 문장 전체의 지배소이고, 따라서 EOS로부터 *gave*로 방향성 링

크(\rightarrow)가 표시되어 있다. 앞으로 $(x \rightarrow y)$ 의 표기는 x 가 y 의 지배소임을 나타낸다.

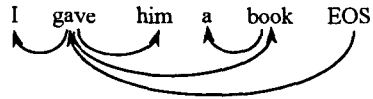


그림 1: 의존관계분석의 예

의존관계분석, \mathcal{D} 는 한 단어열에 대한 구문구조를 나타내며, 다음의 조건을 만족하는 단어간 의존관계의 집합으로 정의된다. (1) 문장전체의 지배소를 제외하고 모든 단어는 문장 안의 임의의 단어를 지배소로 갖는다. (2) 이중 지배소, 의존관계 교차, 의존관계 순환이 없다. (3) 문장전체의 지배소는 EOS의 유일한 피지배소이다.

의존문법에서의 문장확률은 다음과 같이 정의될 수 있다.

$$\begin{aligned} p(w_{1,n}) &= \sum_{\mathcal{D}} p(\mathcal{D}, w_{1,n}) \\ &= \sum_{\mathcal{D}} \prod_{x \rightarrow y \in \mathcal{D}} p(x \rightarrow y), \end{aligned}$$

여기에서

$$\begin{aligned} p(x \rightarrow y) &= p(y|x) \\ &= \frac{\text{freq}(x \rightarrow y)}{\sum_z \text{freq}(x \rightarrow z)}. \end{aligned}$$

2.1 완결-링크와 완결-링크열

여기서, 부분 단어열에 대하여 구문구조의 조건을 만족하는 의존관계 집합에 대한 정의를 필요로 하는데, 이와 같은 단위요소(부분 \mathcal{D})로서 완결-링크와 완결-링크열을 정의한다. 완결-링크와 완결-링크열은 의존문법 구문구조의 기본 구조로서 사용되며, 작은 기본단위로부터 점차 더 큰 완결-링크와 완결-링크열을 형성해 문장 전체에 대한 구조분석, \mathcal{D} 를 형성하게 된다. 또, 다음 섹션에서 단어간 의존관계의 확률값 학습을 위한 기본이 된다.

단어열 $w_{i,j}^1$ 에 대해서 형성되는 모든 의존관계 집합은 다음의 조건들을 만족할 때 완결-링크로 정의된다:

- 배타적으로 $(w_i \rightarrow w_j)$ 혹은 $(w_i \leftarrow w_j)$ 가 존재한다.

¹ w_i, w_j 는 각각 문장내 i 번째, j 번째 단어를 가리키고, $w_{i,j}$ 는 i 에서 $j(i < j)$ 번째까지의 단어열을 가리킨다.

- 내부 단어들은 그 단어열 안에 자신의 지배소인 단어를 갖는다.
- 의존관계의 교차와 순환이 없고, 다중 지배소가 없다.

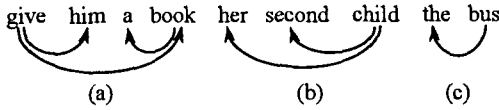


그림 2: 완결-링크 예

완결-링크는 방향성을 가지는데, 가장 바깥 의존관계의 방향에 따라 우향 완결-링크($w_i \rightarrow w_j$ 인 경우) 혹은 좌향 완결-링크($w_i \leftarrow w_j$ 인 경우)가 된다. 최소 단위의 기본 완결-링크는 인접한 두 단어 w_i 와 w_{i+1} 사이의 의존관계로 정의된다. 그림 2에서, (a)는 우향 완결-링크, (b)와 (c)는 좌향 완결-링크의 예이다. 이중 (c)는 기본 완결-링크이기도 하다.

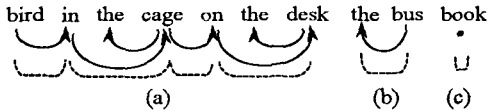


그림 3: 완결-링크열의 예

완결-링크열은 같은 방향성을 가진 0개 혹은 그 이상의 일련의 인접한 완결-링크들로 구성된다. 최소단위 기본 완결-링크열은 한 단어로 구성된 부분 단어열에 정의되는 0개의 연속된 완결-링크이다. 완결-링크열 역시 방향성을 가지며, 구성요소인 완결-링크들의 방향성에 의해 결정된다. 그림 3에서, (a)는 네개의 연속적인 완결-링크로 구성된 우향 완결-링크열, (b)는 하나의 완결-링크로 구성된 좌향 완결-링크열을 나타낸다. (c)는 0개의 완결-링크로 구성된 기본 완결-링크열로서 좌향일 수도 우향일 수도 있다.

“완결”의 의미는 그 단어열 안의 모든 단어에 대한 의존관계는 모두 완결되어 있어서 더 이상의 고려가 필요없다는 의미이다. 앞으로, $w_{i,j}$ 에 대한 우향/좌향 완결-링크는 각각 $L_r(i, j)$ 와 $L_l(i, j)$ 로 표기하고 우향/좌향 완결-링크열은 $S_r(i, j)$ 와 $S_l(i, j)$ 로 표기한다.

임의의 단어열 $w_{i,j}$ 에 대한 완결-링크는 항상 다음과 같은 구조적 결합 특성을 보인다. 임의의 $m(i < m < j)$ 에 대하여

- $L_r(i, j): \{(w_i \rightarrow w_j), S_r(i, m), S_l(m+1, j)\}$
- $L_l(i, j): \{(w_i \leftarrow w_j), S_r(i, m), S_l(m+1, j)\}$

로 구성된다. 그렇지 않으면, 이 완결-링크를 표현하는 링크집합은 링크 교차나 다중 지배소, 혹은 링크-순환의 조건을 만족하지 못하게 되고, 따라서 더이상 완결-링크가 아니게 된다.

마찬가지로 $w_{i,j}$ 에 대한 완결-링크열은 임의의 $m(i < m < j)$ 에 대하여

- $S_r(i, j): \{S_r(i, m), L_r(m, j)\}$
- $S_l(i, j): \{S_l(i, m), L_l(m, j)\}$

로 구성되도록 제한된다. 완결-링크열에 대한 결합 조건을 위와 같이 두는 것은 동일한 완결-링크열의 중복적인 형성을 막기 위함이다².

그림 4는 그림 1의 구문구조를 완결-링크와 완결-링크열의 순차적인 결합구조로 표현한 것이다. 이 중으로 사선 표시된 링크는 완결-링크열을 의미한다. 위에서 아래로 점차 큰 완결-링크와 완결-링크열이 결합되어 마지막의 $S_l(I, EOS)$ 에 의해 하나의 구문트리기가 유일하고 온전하게 표현된다.

그림 5은 임의의 n 개의 단어로 구성된 문장에 대한 추상적인 구문구조, \mathcal{D} 를 나타낸다. $w_k(1 \leq k \leq n)$ 가 문장 전체의 지배소일 때, $(w_k \leftarrow EOS)$ 의 완결-링크가 항상 존재한다는 가정에 의해, 문장 $w_{1,n}$ 의 어떠한 구문구조, \mathcal{D} 도 $S_l(1, EOS)$ 로서 유일하게 표현될 수 있다.

3 학습 알고리즘

본 절에서는 앞 절에서 정의된 완결-링크와 완결-링크열에 대한 내부-외부(Inside-Outside) 확률을 정의하고 이에 기반한 학습 알고리즘을 제시한다. 이 알고리즘은 구문구조문법의 내부-외부(Inside-Outside) 알고리즘[4]의 변형으로 의존 문법에 맞게 수정한 것이다³.

각기 두가지 양상을 나타내는 세가지 요소(내부/외부, 완결-링크/완결-링크열, 우향/좌향)에 의해 8가지의 확률값이 정의된다. 앞으로, β 는 내부 확률값을, α 는 외부 확률값을 의미하고 윗첨자 l 과 s 는 각각 완결-링크와 완결-링크열을 의미한다. 아

²동일한 $S_r(i, j)$ 들이 $L_r(i, m), S_r(m, j)$ 의 결합으로도 형성될 수 있다. $S_l(i, j)$ 의 경우도 마찬가지이다.

³좀더 상세한 수식전개는 [6]에 제시되어 있다.

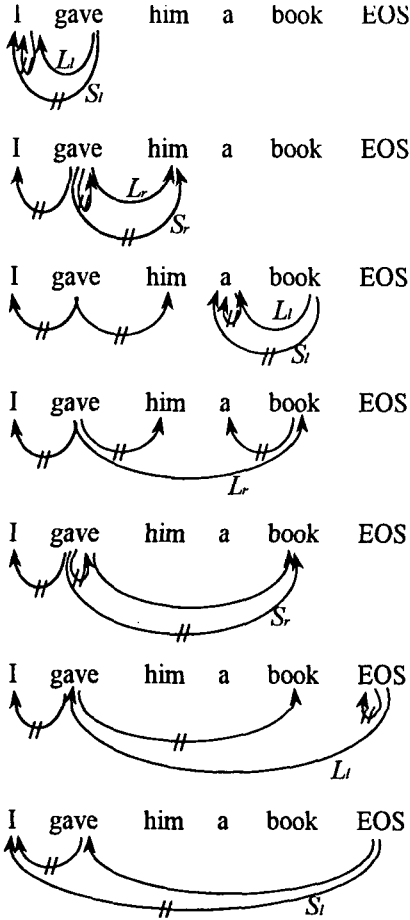


그림 4: 완결-링크와 완결-링크열 구문구조

랫첨자는 방향성을 의미하며, r은 “우향”을, l은 “좌향”을 표시한다.

$L_r(i, j), L_l(i, j), S_r(i, j), S_l(i, j)$ 에 대한 내부 확률은 다음과 같다.

$$\begin{aligned} \beta_r^l(i, j) &= \sum_{m=i}^{j-1} p(w_i \rightarrow w_j) \beta_r^s(i, m) \beta_l^s(m+1, j). \\ \beta_l^l(i, j) &= \sum_{m=i}^{j-1} p(w_i \leftarrow w_j) \beta_r^s(i, m) \beta_l^s(m+1, j). \\ \beta_r^s(i, j) &= \sum_{m=i}^{j-1} \beta_r^s(i, m) \beta_r^l(m, j). \\ \beta_l^s(i, j) &= \sum_{m=i}^{j-1} \beta_l^s(i, m) \beta_l^l(m, j). \end{aligned}$$

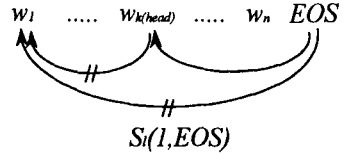


그림 5: D의 추상적 표현

기본 내부 확률값은 다음과 같다.

$$\begin{aligned} \beta_r^l(i, i+1) &= p(w_i \rightarrow w_{i+1}) \\ \beta_l^l(i, i+1) &= p(w_i \leftarrow w_{i+1}) \\ \beta_r^s(i, i) &= \beta_l^s(i, i) = 1 \\ \beta_l^s(1, EOS) &= p(w_{1,n}) \\ \beta_r^s(i, i+1) &= p(L_r(i, i+1)) = p(w_i \rightarrow w_{i+1}) \\ \beta_l^s(i, i+1) &= p(L_l(i, em i+1)) = p(w_i \leftarrow w_{i+1}). \end{aligned}$$

$\beta_l^s(1, EOS)$ 은 문장의 확률로서 그 문장이 가질 수 있는 모든 구문구조 확률의 합이다. 왜냐하면, 모든 의존 구문구조는 $S_l(1, EOS)$ 로 표현되기 때문이다.

$L_r(i, j), L_l(i, j), S_r(i, j), S_l(i, j)$ 에 대한 외부 확률은 다음과 같다.

$$\begin{aligned} \alpha_r^l(i, j) &= \sum_{h=1}^i \alpha_r^s(h, j) \beta_r^s(h, i). \\ \alpha_l^l(i, j) &= \sum_{h=1}^i \alpha_l^s(h, j) \beta_l^s(h, i). \\ \alpha_r^s(i, j) &= \sum_{k=j+1}^n [\alpha_r^s(i, k) \beta_r^l(j, k) \\ &+ \alpha_r^l(i, k) \beta_r^s(j+1, k) Pr(w_i \rightarrow w_k) \\ &+ \alpha_l^l(i, k) \beta_r^s(j+1, k) Pr(w_i \leftarrow w_k)]. \\ \alpha_l^s(i, j) &= \sum_{k=j+1}^n \alpha_l^s(i, k) \beta_l^l(j, k) \\ &+ \sum_{h=1}^{i-1} [\alpha_r^l(h, j) \beta_r^s(h, i-1) Pr(w_h \rightarrow w_j) \\ &+ \alpha_l^l(h, j) \beta_r^s(h, i-1) Pr(w_h \leftarrow w_j)]. \end{aligned}$$

기본 외부확률은 다음과 같다.

$$\alpha_l^s(1, EOS) = 1.$$

내부 확률 ($\beta_r^l, \beta_l^l, \beta_r^s, \beta_l^s$) 계산은 CYK 차트의 관점에서, 상향식(bottom-up)으로, 좌에서 우로 계산된다. 외부 확률들 ($\alpha_r^l, \alpha_l^l, \alpha_r^s, \alpha_l^s$)은 하향식(top-down)으로, 우에서 좌로 계산된다. 이때, 미리 계산된 내부 확률값을 이용한다.

학습 코퍼스가 주어지면, 초기 문법은 학습 코퍼스 내의 모든 유일한 단어들의 쌍으로 초기화 되고, 초기 확률값은 랜덤하게 주어질 수 있다. 이 초기 쌍들은 임시의 단어간 의존관계를 의미하며, 학습 알고리즘이 진행됨에 따라 그 확률값이 조정되어 최종적으로 의미있는 쌍들이 안정적인 확률값과 함께 의존문법으로 정착되게 된다. 학습은 이 초기 문법과 함께 시작되어 학습 코퍼스를 주어진 문법으로 분석하고 각 의존관계의 발생 빈도를 계산한다. 그 발생 빈도를 기반으로 의존관계 확률이 다음과 같이 재조정된다.

$$p_c(w_p \rightarrow w_q) = \frac{C(w_p \rightarrow w_q)}{\sum_{w_r} C(w_p \rightarrow w_r)}$$

이런 코퍼스 분석, 문법 조정의 일련 과정은 학습 코퍼스의 엔트로피가 최소가 될때까지 반복적으로 실행된다. 의존관계, $C(w_i \rightarrow w_j)$ 의 발생빈도는 다음과 같이 계산된다.

$$\begin{aligned} C(w_i \rightarrow w_j) &= \sum_{\mathcal{D}} p(\mathcal{D}|w_{1,n}) O_{cc}(w_i \rightarrow w_j, \mathcal{D}, w_{1,n}) \\ &= \frac{1}{p(w_{1,n})} \alpha_r^l(i, j) \beta_r^l(i, j) \end{aligned}$$

여기에서 $O_{cc}(w_i \rightarrow w_j, \mathcal{D}, w_{1,n})$ 는 만일 의존관계, $(w_i \rightarrow w_j)$ 가 \mathcal{D} 에 사용되었으면 1, 아니면 0의 값을 갖는다. 마찬가지로, 의존관계, $(w_i \leftarrow w_j)$ 의 발생빈도는 $\frac{1}{p(w_{1,n})} \alpha_l^l(i, j) \beta_l^l(i, j)$ 로 계산된다.

4 실험 결과

실험대상으로 KAIST코퍼스⁴에서 추출한 한국어 원형코퍼스를 선택하였으며 이것은 총 13,139 단어와 총 1,589 문장으로 구성되어 있다. 도메인상의 유사성이 유지될 수 있도록 한 분야의 내용으로 제한하였으며 자연환경에서의 동물의 습성에 관한 묘사로 이루어져 있다. 전체 코퍼스에서 144문장을 임의로 뽑아 실험 문장집합으로 선정하고 나머

⁴KAIST(Korea Advanced Institute of Science and Technology)코퍼스는 1994년 이래 구축되어 오고 있으며, 원형 코퍼스(4천 5백만 단어), 품사부착 코퍼스(6백 7십 5만 단어), 트리부착 코퍼스(3만 문장)로 구성되어 있다.

지 1,445문장을 코퍼스의 첫부분부터 100문장씩 증가시키며 잘라서 전부 15개의 학습 코퍼스를 만들고, 이 각각에 대해 학습 및 실험을 하였다.

실험은 세가지 모델을 비교하는 것으로 이루어졌다. 트라이그램에 기반한 모델 (TRI), 바이그램에 기반한 모델 (BI), 그리고 본 논문에서 제시하는 학습된 확률의존문법에 기반한 모델(DEP)을 비교하였다. TRI 모델과 BI 모델의 경우 학습 코퍼스로부터 각 n-gram의 발생 빈도 계산을 통해 학습되었고, DEP 모델의 경우는 제시된 의존문법 학습 알고리즘을 반복적으로 실행함으로써 최적의 문법이 학습되도록 하였다. 그 결과, 각 학습 코퍼스에 대해 보통 26회의 알고리즘 반복을 거쳐 최적의 의존문법이 학습되는 것을 보였다.

언어모델링 실험을 위해서는 평탄화 기법이 필요하다. 평탄화는 학습데이터에서 발견되지 않은 사건의 확률값이 0이 되는 것을 막기위한 것이고, 또 임의의 확률값이 지나치게 과하게 매겨지거나, 헛하게 매겨지는 일을 보장하는 것이다. 평탄화 기법은 그 자체로서 매우 중요한 사항이고 매우 다양한 연구가 진행되고 있지만, 본 실험의 목적은 최적의 평탄화 기법을 찾는 것이 아니기 때문에 하나의 평탄화 기법을 고정시킨 상태에서 모델간의 성능을 비교하였다. 사용된 평탄화 기법은 보간(interpolation)기법의 일종으로 다음과 같다[7].

$$\begin{aligned} \hat{P}_n(w_i|w_{i-n+1}, \dots, w_{i-1}) &= \\ &\lambda P_n(w_i|w_{i-n+1}, \dots, w_{i-1}) + \\ &(1 - \lambda) \hat{P}_{n-1}(w_i|w_{i-n+2}, \dots, w_{i-1}) \end{aligned}$$

평탄화 계수는 발생빈도에 따른 가중치 계수 결정 기법으로 임의의 n-gram의 발생빈도에 따라 계수 값을 달리 정하는 방법으로 하였다.

$$\lambda = \frac{C(w_1, \dots, w_{n-1})}{C(w_1, \dots, w_{n-1}) + K_s}$$

즉, 같은 order의 n-gram에 대해서 같은 가중치 계수를 부여하지 않고 각 n-gram의 발생빈도에 따라 빈도수가 높은 것에 상대적으로 높은 가중치를 주는 방법이다. 여기서 K_s 는 전체의 평탄화 계수로서 값이 크면 클수록 평탄화의 정도가 커진다. 본 실험에서는 K_s 값으로 2를 사용하였다.

모델간 성능비교는 교차 엔트로피 값을 이용하였다.

$$\frac{1}{|V|} \sum_{i=1}^s -\log_2 p_m(s_i)$$

(제 10회 한글 및 한국어 정보처리 학술대회)

위에서 $|V|$ 는 코퍼스를 구성하는 총 단어의 개수, S 는 코퍼스를 구성하는 문장의 개수를 의미한다.

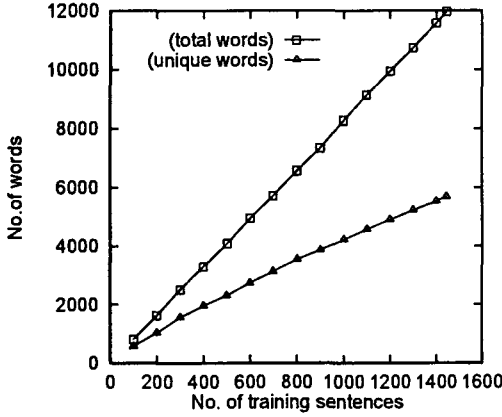


그림 6: 학습코퍼스 증가에 따른 단어 수 증가

그림 6는 실험에서 사용한 학습 코퍼스의 특성을 보이는 것으로 학습 문장수 증가에 대한 전체 단어 수와 유일 단어수의 증가를 보인다. 제한된 영역의, 크기가 큰 코퍼스에 대해 실험을 하면 유일 단어수가 어느 선으로 수렴을 하게될 것이고, 그 경우 각 언어모델링 모델의 비교실험은 더욱 의미있는 결과를 낼 수 있을 것이다.

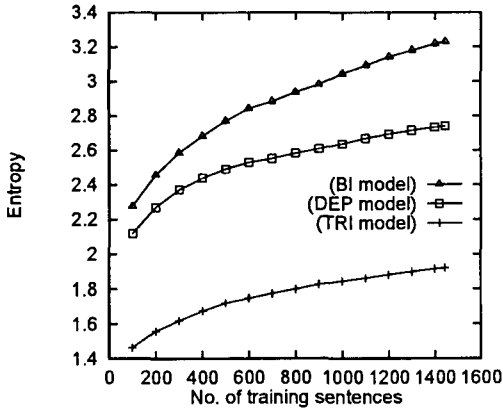


그림 7: 학습코퍼스 교차엔트로피

그림 7은 각 세 모델의 학습코퍼스 엔트로피를 보인다. 학습코퍼스의 경우에는, TRI 모델이 가장 좋은 결과를 보이는 것을 알 수 있다. 이는 다음의 실험 코퍼스에 대한 결과로부터 판단하건대 TRI 모델이 학습코퍼스에 대해 과순응(overfitting) 현상을 보이기 때문이다. DEP 모델은 BI 모델보다 나은 결과를 보이고 있다. 이는 DEP 모델이 지배

소/피지배소의 관계로 표현되는 일종의 언어학적인 bigram 모델이어서, 원거리 의존관계를 표현하는데 더 적합하기 때문이다.

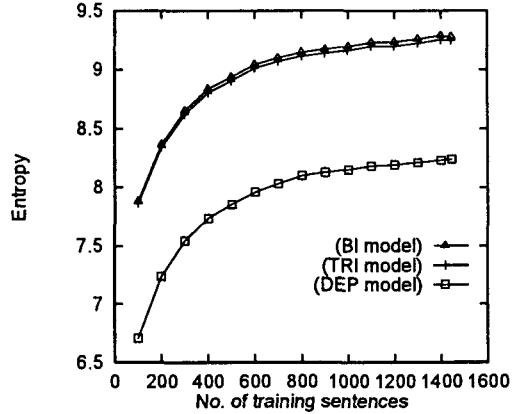


그림 8: 실험코퍼스 교차엔트로피

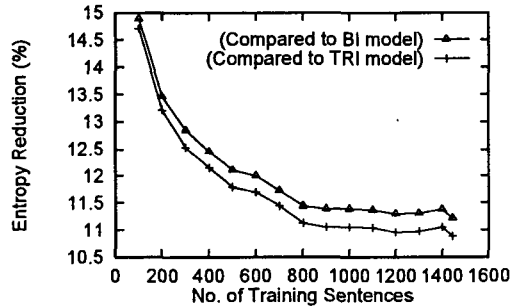


그림 9: 실험코퍼스 교차엔트로피 감소율

실험코퍼스 엔트로피는 그림 8에서 보듯이 BI 모델에 비해 TRI 모델이 약간 좋은 결과를 보였지만 그 차이는 거의 없다. 그러나 의존문법에 의한 DEP 모델은 두 n-gram 모델에 비해 나은 성능을 보였다. 그림 9에서 보듯이 DEP 모델은 최종적으로 BI 모델에 대해서 약 11.5%의 엔트로피 감소율을 보이고, TRI 모델에 대해서 약 11%의 엔트로피 감소율을 보였다.

그림 10는 학습코퍼스와 나머지 실험코퍼스를 통합한 전체에 대한 엔트로피를 실험했을 때의 그래프이다. 실험결과로부터 단어간 의존관계가 단순한 n-gram보다 언어모델링에 더 유용한 정보이고, 의존관계 학습 알고리즘을 통해 원형 코퍼스로부터 숨어있는 단어간 지배-피지배 의존관계를 찾아 낼 수 있음을 알 수 있다.

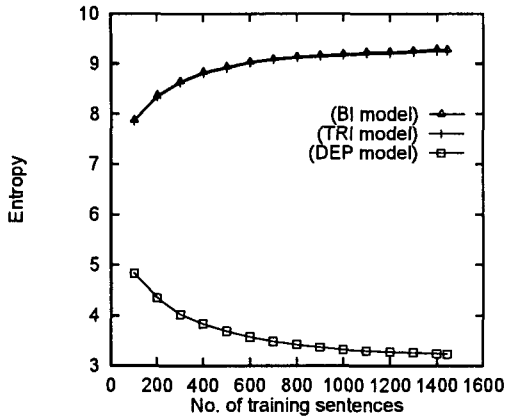


그림 10: 학습/실험 통합코퍼스 교차엔트로피

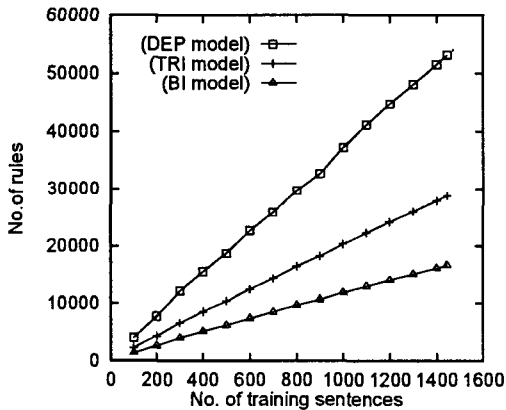


그림 11: 모델 파라미터 크기

그러나, 모델 크기 측면에서는, 그림 11에서 보듯이, DEP 모델이 다른 n-gram 모델들보다 훨씬 많은 파라미터를 필요로 함을 알 수 있다. 이는 방대한 크기의 학습 코퍼스로부터 모델을 구축하고자 할때, 또 모델 크기가 중요한 환경에서는 심각한 문제가 될 수 있다. 하지만, DEP 모델은 학습된 의존문법을 그대로 이용한 것으로써 상당수의 의존관계의 확률값이 0에 가까운 확률값을 가지고 있다. 만일, Seymore와 Rosenfeld [9]가 n-gram 모델에 대해서 실험한 것과 같이, 이렇게 영향이 미미한 의존관계를 제거할 수 있다면, 약간의 엔트로피 상승으로 훨씬 작은 크기의 DEP 모델을 구축할 수 있을 것으로 예상된다.

5 결론

본 논문에서는 단순한 형태의 의존문법에 기반한

언어모델링을 제시하였다. 이 의존문법은 단어간 지배-피지배 관계로 구성되며, 확률의존문법 재추정 알고리즘을 이용하여, 원형 코퍼스로부터 자동으로 학습될 수 있다. n-gram 모델과 비교 실험한 결과, 실험코퍼스에 대해 의존관계 기반 방법이 trigram이나 bigram 모델보다 나은 성능을 보임을 알 수 있었다. 이는 작은 코퍼스에 대해 실험한 결과로서 앞으로, 여러 분야의, 여러 크기의, 그리고 여러 평탄화기법을 가지고 계속 실험을 하면 좀더 의미있는 결론을 내릴 수 있을 것으로 예상된다. 또, 의존관계 기반 방법의 모델 크기가 커지는 문제에 대해서는, 0에 가까운 확률값을 가진 의존관계를 제거하는 기준 및 그 영향에 대한 좀더 세밀한 연구가 진행될 예정이다.

참고문헌

- [1] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. "Class-Based n-gram Models of Natural Language". *Computational Linguistics*, 18(4):467-480, 1992.
- [2] C. Chang and C. Chen. "Application Issues of SA-class Bigram Language Models". *Computer Processing of Oriental Languages*, 10(1):1-15, 1996.
- [3] S. F. Chen. "Building Probabilistic Models for Natural Language". PhD thesis, Harvard University, Cambridge, Massachusetts, 1996.
- [4] F. Jelinek, J. D. Lafferty, and R. L. Mercer. "Basic Methods of Probabilistic Context Free Grammars". Technical report, IBM - T.J. Watson Research Center, 1990.
- [5] K. Lari and S. J. Young. "Applications of stochastic context-free grammars using the inside-outside algorithm". *Computer Speech and Language*, 5:237-257, 1991.
- [6] S. Lee and K. Choi. "Reestimation and Best-First Parsing Algorithm for Probabilistic Dependency Grammar". In *WVLC-5*, pages 11-21, 1997.

- [7] M. K. McCandless. "Automatic Acquisition of Language Models for Speech Recognition". Master's thesis, Massachusetts Institute of Technology, 1994.
- [8] M. Meteer and J.R. Rohlicek. "Statistical Language Modeling Combining N-gram and Context-free Grammars". In *ICASSP-93*, volume II, pages 37-40, January 1993.
- [9] K. Seymore and R. Rosenfeld. "Scalable Trigram Backoff Language Models". Technical Report CMU-CS-96-139, Carnegie Mellon University, 1996.
- [10] S. Sneff. "TINA: A natural language system for spoken language applications". *Computational Linguistics*, 18(1):61-86, 1992.