문장음성인식을 위한 VCCV 기반의 언어모델과 Smoothing 기법 평가

박 선 희[†] · 노 용 완[†] · 홍 광 석[††]

요 약

본 논문에서는 언어모델의 언어처리 단위로 VCCV(vowel consonant consonant vowel) 단위를 제안하고, 기존의 언어처리 단위인 어절, 형태소 단위와 비교한다. 어절과 형태소는 어휘수가 많고 높은 복잡도를 가진다. 그러나 VCCV 단위는 작은 사전과 제한된 어휘를 가지므로 복잡도가 적다. 언어모델 구성에 smoothing은 반드시 필요하다. smoothing 기법은 정확한 확률 예측이 불확실한 데이터가 있을 때 더 나은 확률 예측을 위해 사용된다. 본 논문에서는 형태소, 어절, VCCV 단위에 대해 언어모델을 구성하여 복잡도를 계산하였다. 그 결과 VCCV 단위의 복잡도가 형태소나 어절보다 적게 나오는 것을 볼 수 있었다. 복잡도가 적게 나온 VCCV를 기반으로 N-gram을 구성하고 Katz, Witten-Bell, absolute, modified Kneser-Ney smoothing 등의 방법을 이용한 언어 모델에 대해 평가하였다. 그 결과 VCCV 단위의 언어모델에 적합한 smoothing 기법은 modified Kneser-Ney 방법으로 평가되었다.

Language Model based on VCCV and Test of Smoothing Techniques for Sentence Speech Recognition

Seon-Hee Park[†] · Yong-Wan Roh[†] · Kwang-Seok Hong[††]

ABSTRACT

In this paper, we propose VCCV units as a processing unit of language model and compare them with clauses and morphemes of existing processing units. Clauses and morphemes have many vocabulary and high perplexity. But VCCV units have low perplexity because of the small lexicon and the limited vocabulary. The construction of language models needs an issue of the smoothing. The smoothing technique used to better estimate probabilities when there is an insufficient data to estimate probabilities accurately. This paper made a language model of morphemes, clauses and VCCV units and calculated their perplexity. The perplexity of VCCV units is lower than morphemes and clauses units. We constructed the N-grams of VCCV units with low perplexity and tested the language model using Katz, absolute, modified Kneser-Ney smoothing and so on. In the experiment results, the modified Kneser-Ney smoothing is tested proper smoothing technique for VCCV units.

키워드 : Language Model, Smoothing, VCCV, Perplexity, Speech Recognition

1. 서 론

문장 음성인식에서는 어휘모델, 음향모델, 그리고 언어모델이 인식 성능을 좌우한다. 인식 대상 어휘에 대한 발음정보를 제공해 주는 발음사전을 구성하는 단계가 어휘모델이며 음향 모델은 사람이 발화한 음성으로부터 신호처리와 패턴 매칭을 통해 가능한 발음들을 추정하는 역할을 한다. 그리고 언어 모델의 역할은 음성의 모호함 때문에 음향학적으로 구별하지 못하는 부분을 언어 정보를 이용하여 탐색 공간을 줄이는데 있다. 일반적으로 음성 인식의 성능은 이러한 모델들이 어떻게 결합되어지느냐에 따라 좌우된다[1].

음성인식에 사용되는 언어모델로는 문법기반의 언어모델과 통계적 언어모델이 있으나, 문법기반의 언어 모델은 단어수가 증가하면 network의 상태수가 급격히 증가하여 탐색시간이 오래 걸리는 단점이 있어 일반적으로 검색 시간이 적게 걸리는 통계적 언어모델이 많이 사용되고 있다. 대표적 통계적 언어 모델로는 N-gram을 들수 있으며, 충분한 학습 데이터가 존재할 경우에 매우 좋은 성능을 보여준다[2, 3]. N-gram 언어 모델의 확률값 예측에 중요한 역할을 하는 것은 smoothing이다. smoothing을 통해 불확실한 데이타에 대해 좀 더 정확한 확률값 추정이 가능해진다. smoothing 방법으로는 back off와 interpolation의 두 가지 형태가 있고 각 방식에 적합한 discounting 방법이 있다[4, 5].

언어모델의 언어처리 단위로 한국어에 주로 어절과 형태

※ 본 연구는 2004 정보통신 선도기반기술개발 사업과 한국과학재단 목적기초연구(R05-2002-000-01007-0) 지원으로 수행되었음.
† 준 회 원 : 성균관대학교 대학원 정보통신공학부
†† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2003년 9월 8일, 심사완료 : 2004년 2월 17일

소를 사용한다. 어절은 한국어의 띄어쓰기 기본 단위로 발성의 지속시간이 길기 때문에 N-gram에서 어절 단위로 인식하는 경우 많은 양의 학습 데이터가 존재해야 좋은 성능을 나타낼 수 있다. 형태소를 사용하게 되면 단음소나 단음절을 가진 형태소가 많아서 인식 오류가 증가하게 되므로 인식 성능 개선을 위해 적절한 길이의 발성시간과 적절한 수의 사전 표제어를 가질 수 있게 하는 형태소 결합이 필요하게 된다[6].

본 논문에서는 어절이나 형태소단위보다 모델 구성에 있어서 학습 데이터를 많이 필요로 하지 않고 코퍼스의 크기가 줄어 복잡도가 적어 인식 성능에 좋은 영향을 미치는 VCCV 단위를 언어 처리 단위로 제안한다. VCCV 단위는 인식할 때 VCCV 단위의 음향 모델을 사용할 경우 transcriptiion이 같기 때문에 별다른 처리 없이 바로 매칭해서 쓸 수 있는 편리한 장점을 가지고 있다.

본 논문에서는 VCCV 단위가 한국어 언어모델의 언어처리 단위로 적합한지 알아보기 위해 기존의 언어처리 단위인 어절과 형태소 그리고 VCCV 단위로 언어 모델을 구성하여 어휘수와 복잡도를 비교하였다. 또한 VCCV 단위에 적합한 smoothing 기법이 무엇인지 평가하기 위하여 back off나 interpolation 형태를 가지는 Katz, absolute, Knesar-Ney 등의 smoothing을 사용한 VCCV 단위의 언어 모델을 구성하여 성능 평가를 수행하였다. 본 논문에 사용한 corpus는 스무고개 관련 동물 1764문장으로 그 문장을 training set과 test set으로 나누어 실험에 사용하였다.

2. 언어 모델

언어 모델은 문장 음성을 인식할 때 탐색해야 할 단어의 수를 줄임으로써 문장 구성을 위한 탐색 시간과 인식률을 높이는 역할을 한다[2]. 음성 인식에 주로 사용되는 언어 모델로는 구 구조 문법에 기반한 모델과 통계적 언어 모델을 들 수 있다. 구 구조 문법의 경우 정규 문법이나 문맥 자유 문법을 사용하여 문장의 탐색 시간과 동시에 parsing을 수행하여 문법의 구조에 어긋난 탐색 공간을 제어하게 된다. 비교적 단순한 문법의 경우 FSN을 사용하여 쉽게 표현이 가능한 장점을 가지고 있지만 단어 수가 늘어 나면 네트워크의 상태수가 급격히 증가하게 되어 탐색 시간이 오래 걸리는 단점이 있다. 이해 비해 통계적 언어 모델은 보통 주어진 영역이 많은 텍스트 문장으로부터 쉽게 추출이 가능하고, 입력 문장 전체를 parsing하지 않고 문장의 발생 확률만을 계산하므로 학습 문장과 부분적으로 다른 문장도 인식 할 수 있는 장점이 있다[2, 3, 5].

2.1 통계적 언어 모델

최근에는 가장 그럴듯한 문자열에 가장 높은 확률을 가지도록 정보를 주는 통계적 언어모델이 널리 이용되고 있다. 통계적 언어 모델은 음성인식에서 정확성 뿐만 아니라 탐색 공간을 급격히 줄이는 효과를 보여준다[1]. 통계적 언어 모델의 목적은 주어진 인식 영역에 맞는 단어열 W의 확률을 예측하는 것이다.

단어열 W는 $w_1, w_2, \cdots, w_3$로 이루어진 단어열이라고 가정하면 단어열 W의 확률 P(W)는 식 (2.1)과 같다.

$$p(w_1, w_2, \cdots, w_Q) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2)\cdots p(w_Q|w_1\cdots w_{Q-1}) \quad (2.1)$$

그러나 이 식은 주어진 언어에서 모든 가능한 단어에 대한 P(W)를 계산하기는 용이 하지 않다. 따라서, Markov가정에 의해 P(W)를 근사하는 N-gram을 사용한다. N-gram 방식은 앞의 N-1개의 단어를 바탕으로 현재의 단어에 대한 확률을 계산하는 방법으로 식 (2.2)와 같다.

$$P(w_j|w_1w_2\cdots w_{j-1}) \approx P(w_j|w_{j-N+1}\cdots w_{j-1}) \quad (2.2)$$

일반적으로 N은 2(bigram) 또는 3(trigram) 이외에는 계산이 어렵기 때문에 2나 3이외에는 많이 사용되지 않는다[2]. 언어모델 P(W)는 보통 정해진 영역의 많은 텍스트 코퍼스로부터 추출한다. 텍스트 코퍼스로부터 P(W)를 추정하는 방법은 다음과 같다. 실제적인 관점에서 P(W)는 식 (2.3)과 같이 근사적으로 표시된다.

$$P_N(W) = \prod_{j=0}^{Q} P(w_j|w_{j-1}, w_{j-2}, \cdots, w_{j-N+1}) \quad (2.3)$$

이것을 N-gram laguage model이라고 한다.

조건부 확률 $P(w_j|w_{j-1}, w_{j-2}, \cdots, w_{j-N+1})$는 식 (2.4)에 의해서 추정될 수 있다.

$$\hat{P}(w_j|w_{j-1}, w_{j-2}, \cdots, w_{j-N+1}) = \frac{F(w_j w_{j-1}, \cdots, w_{j-N+1})}{F(w_{j-1}, \cdots, w_{j-N+1})} \quad (2.4)$$

여기에서 F는 훈련 코퍼스에서 발생된 문자열의 전체 수이다. 그런에 이것이 잘 추정되기 위해서는 훈련 코퍼스가 매우 커서 단어열이 충분히 나타나야 한다.

실제로 $F(w_j, w_{j-1}, \cdots, w_{j-N+1}) = 0$이 되는 단어열이 많이 나타난다. 이러한 문제점을 극복하는 하나의 방법은 N이 3이라고 했을 때 trigram, bigram, unigram의 확률을 이용하는 것이다. 식 (2.5)과 같다.

$$\hat{P}(w_j|w_{j-1}, w_{j-2}) = P_1\frac{F(w_1, w_2, w_3)}{F(w_1, w_2)} + P_2\frac{F(w_1, w_2)}{F(w_1)} + P_3\frac{F(w_1)}{\Sigma F(w_1)} \quad (2.5)$$

여기에서 $P_1 + P_2 + P_3 = 1$이고 $\Sigma F(w_j)$는 코퍼스 전체의 크기이다[7, 8].

3. VCCV 단위의 언어 모델

본 장에서는 2장에서 설명한 통계적 언어 모델을 사용하여 제안하는 VCCV 단위로 언어처리를 하는 방법에 대해서 설명한다. 먼저 표준 발음 변환에 대해 설명하고 VCCV 단위에 대한 개념과 모델링 하는 방법 그리고 복잡도에 대해서 설명한다.

3.1 표준 발음 변환

음성 인식을 위해서는 음운 변화가 적용된 인식 단위를 처리해야 하므로 음운 변화가 반영된 발음으로 변환해야 한다[9]. VCCV 단위는 모음의 정보를 이용해서 분할하는 단위이므로 발음 변환에 따라 구성이 변할 수 있으므로 발음 변환이 중요하다. 본 논문에서는 가장 일반적인 표준 발음법을 적용하여 발음 변환을 하였다.

예를 들어, "강에 살고 있습니까"란 문장에 대해서 표준 발음법을 적용하게 되면 '있'의 ㅆ이 ㄷ으로 바뀌고 '습'의 ㅂ이 ㅁ으로 바뀌어서 "강에 살고 읻슴니까"로 바뀌게 된다. 본 논문에서 사용한 표준 발음법은 문교부 고시 제 38-2 호에 규정되어 있는 것을 사용하였다.

3.2 VCCV 단위

인식 단위의 경계 추출은 무성음과 유성음 사이에서 뚜렷하게 나타나는 경우가 있지만 유성음, 무성음 혹은 유성음과 무성음들 사이에서 상호간의 조음현상 때문에 정확한 경계를 찾기가 어려울 뿐만 아니라, 이러한 경계 영역의 천이 구간에서 음성 데이터들은 별다른 의미를 갖지 못한다. 이러한 천이 구간의 데이터는 조음 현상에 의하여 인접 음소의 영향을 받으므로 인접 음소에 따라 특성이 달라지기 때문이다. 따라서 이러한 음소의 경계를 정확히 찾는 것보다 특성의 변화가 적은 안정된 대표구간, 즉 모음 영역을 찾는 것이 더 효율적이라 할 수 있다[9]. 이러한 단위를 음향 모델 뿐만 아니라 언어 모델의 언어 처리 단위로도 적용하기 위해서 텍스트를 VCCV로 분할하여 사용한다. 언어학적 특성인 모음 정보를 이용하여 VCCV 단위로 분할하는 예를 보여주기 위해 <표 1>의 코퍼스를 예로 들었다.

〈표 1〉 Corpus의 예

Corpus
1. 강에 살고 읻슴니까
2. 날개가 읻는 곤충 임니까

<표 1>처럼 코퍼스가 주어졌을 때 코퍼스의 음절 수와 모음 정보를 이용하여 코퍼스를 VCCV 단위로 분할하는데 '강에'는 두 음절로 이루어져 있고, 그중에서 모음은 '아'와 '에'의 두 개로 이루어져 있다. 이것을 VCCV로 나누면 '아'와 '에'를 경계로 '가', '앙에', '에'의 세 부분으로 나누어진다. <표 1>에 코퍼스를 VCCV 단위로 나눈 예가 <표 2>와 같다.

〈표 2〉 VCCV 단위의 Corpus

VCCV 단위의 Corpus
1. 가/CV 앙에/VCV 에/V ㅣ 사/CV 알고/VCCV 오/V ㅣ 이/V 읻스/VCCV 음니/VCCV 이까/VCV 아/V
2. 나/CV 알개/VCCV 애가/VCV 아/V ㅣ 이/V 인느/VCCV 은/VC ㅣ 고/CV 온추/VCCV 웅/VC ㅣ 이/V 임니/VCCV 이까/VCV 아/V

본 논문에서는 코퍼스가 분할되어 나타날 수 있는 단위인 V(vowel), VC(vowel consonant), VCV(vowel consonant vowel), VV(vowel vowel), VCCV(vowel consonant consonant vowel), CV(consonant vowel) 등을 한꺼번에 묶어 VCCV 단위라 한다. 전체 텍스트에서 나올 수 있는 VCCV 단위중에 어절의 앞부분으로 올 수 있는 것의 개수는 305, 어절의 뒷부분에 올 수 있는 개수는 56개이다. 논 논문에서 사용한 스무고개 관련 코퍼스는 어절의 앞부분인 CV 단위로 104개, 어절의 뒷부분인 VC 단위로 40개를 사용하였다.

3.3 VCCV 단위의 언어 모델링

본 논문에서는 코퍼스가 주어지면 그 코퍼스에 대해 발음변환을 한 후 VCCV 단위로 분할한 다음 bigram으로 언어모델을 구성하였다. 구성하는 방법은 첫 번째로 문장이 VCCV로 나누어지면 각 문장의 처음과 끝 기호를 부여한다. 두 번째로 문장에서 VCCV 회수가 bigram 형태인 것의 회수를 센다. 마지막으로 그 회수를 가지고 bigram을 계산한다.

3.4 복잡도(perplexity)

복잡도는 언어모델의 복잡한 정도를 나타내는 것으로 일반적으로 사용하는 언어 모델에서의 객관적인 평가척도를 말한다[10]. 이를 위한 척도로 cross entropy가 일반적으로 이용되고 있다.

언어 L에 대한 모델 M을 생각하고 언어 모델 M에 의한 단어열 $w_1, w_2, \cdots, w_n$의 생성확률을 $P_M(w_1, \cdots, w_n)$으로 나타낸다. 이 때 언어 모델 M에 대한 언어 L의 cross entropy와 복잡도 pp를 식 (3.1)과 같이 정의한다[11].

$$H_0(L,M) = -\sum_{w_1 \cdots w_n} P(w_1 \cdots w_n) \times \log P_M(w_1, \cdots, w_n) \qquad PP = 2^{H_0} \tag{3.1}$$

4. Smoothing

언어모델에서 나올 수 있는 경우가 드문 어휘에 대한 확률의 올바른 추정을 위해서는 smoothing 기술이 필수적이다[5]. Maximum likelihood에 의해서 언어 모델의 확률값을 구하게 되면 zero-count 단어들에 대해서는 zero의 확률값이 주어지게 되고 nonzero-count 단어들에 대해서는 너무 높은 확률값이 주어진다. 이러한 값들에 대해서 높은 확률

값은 discount 해주고 zero 확률값에 대해서도 어느 정도의 확률을 부여해주는 smoothing을 해주어야 한다. 가장 간단한 smoothing 기법 중에 하나는 +1 smoothing으로 각 count에 대해서 1을 더해주는 방법으로 식 (4.1)과 같다[4, 5].

$$P(w_j|w_{j-1}) = \frac{1 + C(w_{j-1}w_j)}{|V| + C(w_{j-1})} \tag{4.1}$$

그러나 zero-count에 대한 확률값이 모두 같고 V(vocabulary)가 너무 크게 되면 확률값 추정이 올바르지 않다[4]. 이러한 문제점을 극복하는 여러 가지 smoothing 기법들을 이 장에서 보여준다.

4.1 Back off and Interpolation

Smoothing 하는 형식에 따라 back off 방법과 interpolation 방법이 있다. Back off 방법은 bigram이 존재하는 경우에는 unigram의 영향을 받지 않고 bigram이 존재 하지 않을 경우에만 unigram의 영향을 받는다. 그러나 interpolation은 bigram을 구할 때 unigram의 영향을 받는다[4].

4.2 Smoothing 기법

본 논문의 실험에 사용한 몇 가지 Smoothing 기법들에 대해서 소개한다. Absolute smoothing은 back off 형태이고 absolute discounting을 사용한다. Bigram 형태는 식 (4.2)와 같다.

$$P_{abs}(w_j|w_{j-1}) = \begin{cases} C\dfrac{(w_jw_{j-1})-D}{C(w_j)} & \text{if } C(w_{j-1}w_j)>0 \\ D\dfrac{N-n_0(w_{j-1})}{C(w_{j-1})}P(w_j) & \textit{otherwise} \end{cases} \tag{4.2}$$

여기서 D는 n_1/n_1+2n_2이고 n_r은 r번 나타난 bigram의 수이다[4, 5].

Katz smoothing은 back off 형태이고 Good-Turing discount을 사용한다. Good-Turing 방식은 bigram이 r번 발생하는 상태는 추정하는 것으로 r^*로 나타내고 r^*는 식 (4.3)과 같다.

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{4.3}$$

Katz smoothing의 bigram 형태는 다음 식 (4.4)와 같고 d_r값은 식 (4.5)와 같다.

$$P_{katz}(w_j|w_{j-1}) = \begin{cases} C(w_{j-1}w_j)/C(w_{j-1}) & \text{if } r>k \\ d_rC(w_{j-1}w_j)/C(w_{j-1}) & \text{if } k\geq r>0 \\ \alpha(w_{j-1})P(w_j) & \text{if } r=0 \end{cases} \tag{4.4}$$

$$d_r = \begin{cases} 1 & r>k \\ \dfrac{\dfrac{r^*}{r}-\dfrac{(k+1)n_{k+1}}{n_1}}{1-\dfrac{(k+1)n_{k+1}}{n_1}} & r<k \end{cases} \tag{4.5}$$

$\alpha(w_{j-1})$은 확률값의 합이 1이 되어야 하기 때문에 식 (4.6)으로 계산된다[5, 13].

$$\alpha(w_{j-1}) = \frac{1-\Sigma_{w_j:r>0}P_{katz}(w_j|w_{j-1})}{1-\Sigma_{w_j:r>0}P(w_j)} \tag{4.6}$$

Kneser-Ney smoothing은 back off 형태이고 absolute discount을 사용한다. bigram 형태는 다음 식 (4.7)과 같고 $P_{KN}(w_j)$와 $\alpha(w_{j-1})$은 식 (4.8)과 같다[5, 12].

$$P_{KN}(w_j|w_{j-1}) = \begin{cases} \dfrac{C(w_jw_{j-1})-D}{C(w_j)} & \text{if } C(w_{j-1}w_j)>0 \\ \alpha(w_{j-1})P_{KN}(w_j) & \textit{otherwise} \end{cases} \tag{4.7}$$

$$P_{KN}(w_j) = \frac{C(\cdot\, w_j)}{\Sigma_{w_j}C(\cdot\, w_j)},$$

$$\alpha(w_{j-1}) = \frac{1-\Sigma_{w_j:C(w_{j-1}w_j)>0}\dfrac{C(w_{j-1}w_j)-D}{C(w_{j-1})}}{1-\Sigma_{w_j:C(w_{j-1}w_j)>0}P_{KN}(w_i)} \tag{4.8}$$

식 (4.8)에서 $C(\cdot\, w_j)$는 w_j에 선행하는 unique한 단어의 수이다.

Witten-Bell smoothing은 interpolated 형태이고 linear discount을 사용한다. 이 기법은 bigram의 count수에 의존하지 않고 t에 따라서 discounting된다. 이 smoothing 기법의 bigram 형태는 식 (4.9)와 같다.

$$P_{WB}(w_j|w_{j-1}) = \frac{C(w_{j-1})}{C(w_{j-1})+t}\frac{C(w_{j-1}w_j)}{C(w_{j-1})} + \frac{t}{C(w_{j-1})+t}P(w_j) \tag{4.9}$$

여기서 t는 특정 문맥 다음에 올수 있는 다른 event의 수이다[7].

Modified Kneser-Ney smoothing 기법은 Knerser-Ney의 변형된 형태로 back off 대신에 interpolated 형태를 사용하고 하나의 absolute discount가 아닌 3개의 absolute discount을 사용한다. 식은 다음 식 (4.10)~식 (4.13)과 같다[4, 5].

$$P_{M-KN}(w_j|w_{j-1}) = \frac{C(w_{j-1}w_j)-D(C(w_{j-1}w_j))}{C(w_{j-1})} + \gamma(w_{j-1})P(w_j) \tag{4.10}$$

$$D(c) = \begin{cases} 0 & \text{if } c=0 \\ D_1 & \text{if } c=1 \\ D_2 & \text{if } c=2 \\ D_{3+} & \text{if } c\geq 3 \end{cases} \tag{4.11}$$

$$\gamma(w_{j-1}) = \frac{D_1n_1(w_{j-1}\cdot)+D_2n_2(w_{j-1}\cdot)+D_3n_3(w_{j-1}\cdot)}{C(w_{j-1})} \tag{4.12}$$

$$D_1 = 1-2Y\frac{n_2}{n_1},\quad D_2 = 2-3Y\frac{n_3}{n_2},\quad D_{3+} = 3-4Y\frac{n_4}{n_3}$$

$$\left(Y=\frac{n_1}{n_1+2n_2}\right) \tag{4.13}$$

Jelinek-Mercer smoothing 기법은 interpolated 형태이고 linear discount를 사용한다. 그 식은 다음 식 (4.14)와 같다 [4, 12].

$$P_{JM}(w_{j-1}|w_j) = \lambda \frac{C(w_{j-1}w_j)}{C(w_{j-1})} + (1-\lambda)P(w_j)$$
$$\lambda = n_1/N \qquad (4.14)$$

λ는 Good-Turing discount에서 나타나지 않은 이벤트와 같은 확률값이다.

4.3 언어모델의 평가

언어모델을 평가하는 가장 일반적인 방법은 test data에 할당된 모델의 확률값을 측정하거나 cross-entropy와 perplexity 값을 측정하는 것이다. 낮은 perplexity는 일반적으로 낮은 word error rate를 갖기 때문에 인식 성능에 중요한 영향을 미치게 된다. smooth된 n-gram모델이 확률값 $P(w_j|w_{j-1})$를 가지면 식 (2.3)을 사용하여 문장에 대한 확률값을 계산할 수 있다. 문장 $(t_1, \cdots, t_{l_T})$을 가지는 test 집합 T에 대한 확률값 P(T)는 다음 식 (4.15)처럼 test 집합에 있는 모든 문장의 확률값의 곱으로 나타낼 수 있다[4, 5].

$$P(T) = \prod_{i=1}^{t_r} p(t_i) \qquad (4.15)$$

이것에 대한 cross entropy $H_p(t)$는 식 (4.16)과 같다.

$$H_{p(T)} = -\frac{1}{W_T} \log_2 P(T) \qquad (4.16)$$

W_T는 w로 측정한 text T의 길이이다.

Perplexity $PP_p(T)$는 $2^{H_p(T)}$이다. 낮은 perplexity를 가진 언어 모델이 좋은 성능을 가진다[5, 12, 13].

5. 실험 및 결과

본 논문에서 사용한 DB는 스무고개 게임에 적용하기 위해서 스무고개에서 나올 수 있는 질문 중에 동물 영역의 데이터를 가지고 구성하였다. 전체 문장은 1746문장이고 그 중에서 smoothing 할 때 training에 사용된 문장은 1624문장이고 test에 사용한 문장은 122문장이다. 1746문장에 대한 어절, VCCV, 형태소의 개수는 <표 3>과 같다.

〈표 3〉 어절, VCCV, 형태소 단위의 개수

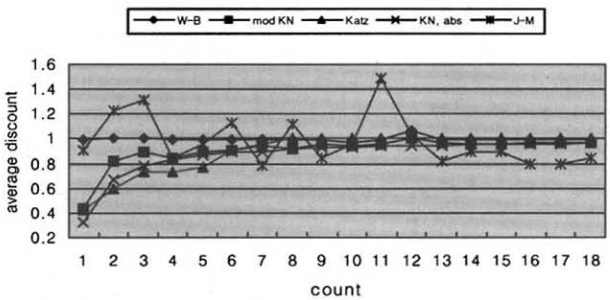
문 장		1746
어 절		1955
VCCV	CV	154
	VCCV	1404
	VC	40
형 태 소		1543

1746문장에 대한 어절과 형태소 그리고 VCCV 단위로 bigram 언어 모델을 구성하고 각각의 복잡도를 계산한 결과는 <표 4>와 같다.

〈표 4〉 어절, 형태소, VCCV 단위의 복잡도

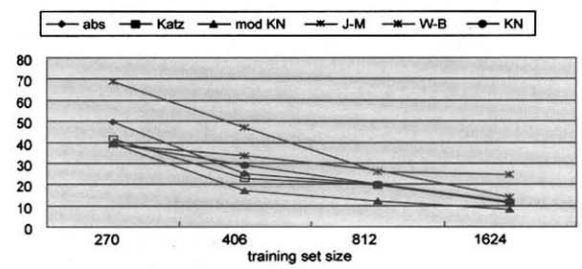
단 위	복 잡 도
어 절	3.98
형 태 소	2.57
VCCV	2.35

<표 4>의 결과에서 VCCV 단위가 복잡도가 가장 적게 나오는 것을 볼 수 있다. VCCV 언어모델을 각 smoothing 방법에 따라 smooth하고 discount 정도를 보기 위해 count 1-17까지의 평균 discount된 확률값을 구한 결과는 (그림 1)과 같다. 그림에서 good-turing discount나 absolute discount를 사용한 Katz, abs, KN, mod KN 방식은 bigram의 count수에 비례하여 count수가 적을수록 더 많이 discount되는 것을 볼수 있다.



(그림 1) bigram count에 따른 평균 discount된 확률값의 비

Training set 크기에 따른 smooth된 언어 모델의 성능을 평가하기 위해 trainging set 크기 270문장, 406문장, 812문장, 1624문장에 대한 test set의 perplexity를 계산하였다. 그 결과 (그림 2)에서 볼 수 있듯이 training set 크기가 클수록 복잡도가 적게 나타났다.



(그림 2) training set size에 따른 perplexity

(그림 2)에서 성능이 가장 좋게 나타나는 1624문장의 training set에 대한 perplexity를 smoothing 방법에 따라 기록한 것이 <표 5>와 같다.

〈표 5〉 Smoothing 방법에 따른 perplexity

smoothing 기법	Perplexcity
absolute	11.57
Katz	10.99
Kneser-Ney	11.46
Witten-Bell	24.49
modified Kneser-Ney	8.18
Jelinek-Mercer	19.93

1624문장으로 training한 data에 대한 test data의perplexity를 계산한 결과 modified Kneser-Ney smoothing이 가장 낮게 나왔다. 그러므로 본 논문에서 사용한 corpus에 대해서는 Kneser-Ney smoothing을 사용한 언어 모델 방법이 효율적인 것을 볼수 있다. Witten-Bell, Jelinek-Mercer smoothing 방법은 효율성이 많이 떨어지는 것을 볼 수 있다.

6. 결 론

본 논문에서는 VCCV 단위로 언어 모델을 구성하여 기존의 언어 모델 언어 처리 단위와 비교하였다. bigram 언어모델을 적용하여 코퍼스를 형태소, 어절, VCCV로 나누어서 실험한 결과 VCCV가 어휘수도 적게 나오면서 복잡도가 적게 나온 것을 볼 수 있었다. 복잡도가 적게 VCCV 언어 모델의 올바른 확률값 추정을 위해서 smoothing 기법을 적용하여 언어 모델의 성능을 평가하였다. 그 결과 언어 모델의 성능이 가장 좋게 나온 smoothing 방법이 Modified Kneser-Ney로 평가되었다. 향후 연구 과제로는 이러한 smoothing 기법을 가진 VCCV 언어 모델을 문장 음성인식에 적용하여 인식 성능을 평가하고 다른 단위와 비교 할 것이다.

참 고 문 헌

[1] R. lyer and M. Ostendorf, "Relevance weighting for combining multi-domain data for n-gram language modeling," Computer Speech and Language, 13, pp.267-282, 1999.

[2] 오영환, "음성언어정보처리", 홍릉과학출판사, 1997.

[3] 이건상, 양성일, 권성헌 공저, "음성인식", 한양대학교 출판부, 2001.

[4] Stanley F. Chen and Joshua Goodman, "An Emperical Study of Smoothing Techniques for language modeling," Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

[5] Huang X., Acero A., Hon H.-W., "Spoken language processing," Prentice Hall PTR, October, 2001.

[6] Laurence Rabiner and Bing-Hwang Jang, "Fundamantals of Speech Recognition," Printice-Hall EngleWood Cliffs, NJ., 1993.

[7] P. R. Clarkson and R. Rosenfeld, "Statistical Language Modeling using The CMU-Cambridge Toolkit," ESCA Eurospeech, 1997.

[8] 이진석, 박재득, 이근배, "K-SLM toolkit을 이용한 한국어의 통계적 언어 모델링 비교", 한국전자통신연구원, 1999.

[9] 윤재선, "한국어 음성 인식 dictation system의 구현", 성균관대학교 정보통신공학과 박사학위논문, 2001.

[10] 남지순 "한국어 전자사전" 전자공학회지, 제24권 제9호 pp. 1103-1125, Sep., 1997.

[11] 北 研二, "音聲言語處理", 森北出版株式會社, 1998.

[12] Jelinek and Frederick, "Statistical Methods for Speech Recoginition," MIT press, 1997.

[13] Steve Young and Gerrit Bloothooft, "Corpus-Based Methods in Language andspeech processing," Kluwer Academic Publishers, 1997.

박 선 희

e-mail : seonhp@hotmail.com

2002년 호남대학교 정보통신공학과(학사)

2004년 성균관대학교 정보통신공학부 (공학석사)

관심분야 : 음성인식, 언어처리, 음성이해

노 용 완

e-mail : elec1004@hotmail.com

2001년 남서울대학교 정보통신공학과(학사)

2003년 성균관대학교 정보통신공학부 (공학석사)

2003년~현재 성균관대학교 정보통신공학부 박사과정

관심분야 : 음성인식, 음성이해, 신호처리

홍 광 석

e-mail : kshong@skku.ac.kr

1985년 성균관대학교 전자공학과(학사)

1988년 성균관대학교 전자공학과(공학석사)

1992년 성균관대학교 전자공학과(공학박사)

1990년~1993년 서울보건전문대학 전산정보처리과 전임강사

1993년~1995년 제주대학교 정보공학과 전임강사

1995년~현재 성균관대학교 정보통신공학부 부교수

관심분야 : 음성인식 및 합성, HCI