말소리와 음성과학 제6권 제3호 (2014.09.30)
ISSN 2005-8063, pp. 155~164
www.speechsciences.or.kr
http://dx.doi.org/10.13064/KSSS.2014.6.3.155

비교사 분할 및 병합으로 구한 의사형태소 음성인식 단위의 성능

Performance of Pseudomorpheme-Based Speech Recognition Units Obtained by Unsupervised Segmentation and Merging

방 정 욱[1] · 권 오 욱[2]
Bang, Jeong-Uk · Kwon, Oh-Wook

ABSTRACT

This paper proposes a new method to determine the recognition units for large vocabulary continuous speech recognition (LVCSR) in Korean by applying unsupervised segmentation and merging. In the proposed method, a text sentence is segmented into morphemes and position information is added to morphemes. Then submorpheme units are obtained by splitting the morpheme units through the maximization of posterior probability terms. The posterior probability terms are computed from the morpheme frequency distribution, the morpheme length distribution, and the morpheme frequency-of-frequency distribution. Finally, the recognition units are obtained by sequentially merging the submorpheme pair with the highest frequency. Computer experiments are conducted using a Korean LVCSR with a 100k word vocabulary and a trigram language model obtained by a 300 million eojeol (word phrase) corpus. The proposed method is shown to reduce the out-of-vocabulary rate to 1.8% and reduce the syllable error rate relatively by 14.0%.

Keywords: Pseudomorpheme, Korean LVCSR

1. 서론

한국어 대어휘 연속음성인식(large vocabulary continuous speech recognition; LVCSR)을 위한 인식단위로는 음소, 음절, 형태소, 어절이 가능하다[1][2][3]. 음소 단위 인식기의 경우 어휘 개수는 음소 개수와 같으므로 인식과정은 단순하나, 인식단위의 평균 지속시간이 짧고, 음소간의 언어모델이 적용되므로 인식률이 저하된다. 어절 단위 인식기의 경우 말뭉치의 양이 제한되므로 강인한 언어모델은 구하기가 어렵지만, 인식단위의 평균 지속시간이 길어지므로 탐색기에서 넓은 범위의 문맥을 고려할 수 있다. 그러나 모든 종류의 어절을 인식 어휘에 넣어야 하므로 탐색공간이 증가하고, 어휘외 (out-of-vocabulary; OOV) 단어가 증가하므로 대어휘 연속음성인식기의 인식단위로는 적합하지 않다. 형태소 단위는 음절 단위보다 평균 지속시간이 길면서 어절 단위처럼 문맥을 고려할 수 있다. 그러나 'ㄴ', 'ㄹ', '이' 등과 같은 단음소와 대부분의 의존명사 또는 접미사가 단음절로 존재할 수 있으며, 이러한 형태소는 매우 짧은 시간 동안에 발성되기 때문에 이를 인식하기에는 많은 어려움이 있다.

한국어 텍스트로부터 인식단위를 구하는 방법은 형태소 분할과 비교사(unsupervised) 분할 방법이 있다. 형태소 분할 방법[1][2][3]은 형태소 분석기를 이용하여 한국어 문장을 형태소 단위로 분할하고, 빈도가 높은 형태소를 병합하여 인식단위로 사용한다. 하지만, 고유명사 또는 신조어와 같은 새로운 단어가 생길 때마다 형태소 사전 목록에 추가해야 하는 불편함이 있다.

이와는 대조적으로 비교사 분할 방법은 미리 정의된 규칙을 따르지 않고 단어의 빈도나 길이 정보를 이용하여 새로운 단위를 만든다[4]. 이 경우 언어학적 정보를 필요로 하지 않아 여러 언어에서 공통적으로 사용될 수 있지만, 경우에 따라 적절한 파라미터 설정이 필요하며, 많은 양의 말뭉치를 필요로

1) 충북대학교, isaac@cbnu.ac.kr
2) 충북대학교, owkwon@cbnu.ac.kr, 교신저자
이 논문은 2012년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

접수일자: 2014년 7월 29일
수정일자: 2014년 8월 26일
게재결정: 2014년 9월 10일

한다.

본 논문에서는 형태소 분할 방법을 이용한 인식단위에서 고유명사 또는 형태소 분석 어휘 부족에서 기인하는 OOV 단어를 감소시키기 위하여, 1단계에서는 비교사 분할 방법을 적용하여 기존의 형태소 인식단위를 더 작은 부형태소(submorpheme) 단위로 분할하고, 2단계에서 한 어절 범위 내에 속하는 단위 중에서 발생 빈도가 높은 쌍을 병합하여 의사형태소 인식단위를 생성하는 방법을 제안한다. 제안한 방법은 복잡도(perplexity; PP)를 크게 증가시키지 않으면서도 인식단위의 평균 지속시간을 증가시킴으로써 오류율을 감소하는 효과를 갖는다.

본 논문의 구성은 다음과 같다. 2장에서는 LVCSR의 인식단위를 구하기 위한 기존의 형태소 분할 방법과 비교사 분할 및 병합 방법을 소개하고, 3장에서는 본 연구에서 제안한 형태소 분할, 비교사 분할, 병합 과정을 결합한 방법을 설명하고, 4장에서는 인식단위 실험 결과를 보여주고, 5장에서는 음성인식 실험 결과를 보여주고, 6장에서 결론을 맺는다.

2. 기존 방법

2.1 형태소 분할 방법

형태소는 음성언어에서 의미를 가지는 가장 작은 요소이다[5]. 형태소 단위는 <그림 1>과 같이 형태소 분석기를 이용하여 생성된다. 먼저 전처리를 통해 문장 부호나 특수기호가 제거된 어절 단위 말뭉치를 미리 정의된 형태소 사전과 형태소 확률모델을 사용하여 문법 형태소를 분리하고, 체언 및 용언 분석, 단일 형태소 분석을 한다. 마지막으로 불규칙 사전을 참고하여 최종적인 형태소 단위를 생성한다[5].

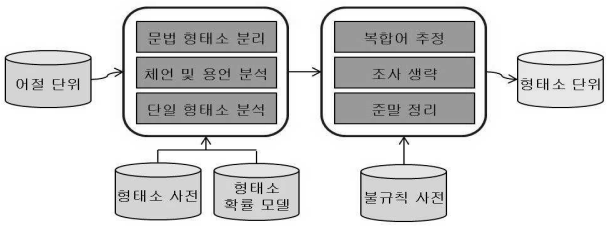

그림 1. 한국어 형태소 분석기의 구조
Figure 1. Architecture of a Korean morphological analyzer

한국어 형태소 분석에서 가장 해결하기 어려운 부분이 복합어와 미등록어(형태소 사전에 등록되지 않은 형태소) 처리이다. 이외의 어절은 규칙에 따라 처리할 수 있지만, 미등록어와 복합어의 경우 명확하게 인식할 수 있는 방법이 없다. 때문에 이러한 복합어나 미등록어를 얼마나 잘 처리하느냐가 형태소 분석기의 성능에 중요한 평가 요소가 된다[5].

일반적인 형태소 분석기의 결과는 형태소 분할 과정에서 발음열의 변화가 생길 수 있기 때문에, 음성인식에 사용되기가 어렵다. 따라서 일반 형태소 분석기를 수정한 의사형태소(pseudomorpheme) [1] 분석기를 이용하여 얻어지는 발음이 유지되는 의사형태소 단위가 주로 사용된다. 하지만, 의사형태소를 그대로 음성인식에 사용하는 경우 단음소와 단음절로 이루어진 의사형태소에 의하여 인식률 저하가 발생하는데, 이를 개선하고자 자주 발생하는 의사형태소 쌍을 병합하여 인식 단위로 사용한다. 이렇게 병합된 의사형태소 단위는 평균 지속시간이 증가되어 인식률 향상에 기여한다[3]. 이후로는 의사형태소를 편의상 “형태소”로 부르기로 하며, 혼동이 발생할 수 있는 부분에서는 구분하여 사용하도록 한다.

2.2. 비교사 분할 및 병합 방법

비교사 분할 단위의 경우 언어학적 정보가 사용되지 않고 단어의 빈도나 길이 분포를 이용하여 새로운 단위를 생성한다. 비교사 분할 방법 중 하나인 Morfessor [6]는 어절 단위의 입력에서 언어학적 정보를 사용하지 않고 형태소 단위와 유사한 결과 생성을 목적으로 하는 알고리듬이며, word piece model [7]은 음절 단위의 빈도 정보를 이용하여 병합하여 확장하는 알고리듬이다.

2.2.1. Morfessor

Morfessor는 문맥 독립(context-independent)인 Morfessor Baseline과 문맥 종속(context-dependent)인 Categories-ML, Categories-MAP로 분류된다[8]. 본 논문에서는 3가지 버전 중에서 한국어와 비슷한 형태론적 특징을 가진 핀란드어에서 가장 높은 인식 성능을 가지는 Morfessor Baseline[8]을 선정하고 이를 설명한다.

Morfessor는 단어의 사후확률(posterior probability)을 최대화하는 단위 경계를 재귀적으로 찾아 분할하는 알고리듬이다. 분할에 사용되는 통계적인 정보로는 단어 빈도(word frequency), 문자 빈도, 단어 길이 분포, 단어 빈도의 빈도[3] (frequency of frequency; fof)이다. 사후확률은 수식 (1), (2), (3)과 같이 정의된다[9].

$$P(C|M) = \prod_{j=1}^{W}\prod_{k=1}^{n_j} P(\mu_{jk}) \tag{1}$$

$$P(M) = N!\, P(f_{\mu_1},\ldots,f_{\mu_M})P(s_{\mu_1},\ldots,s_{\mu_M}) \tag{2}$$

$$\arg\max_M P(M|C) = \arg\max P(C|M)P(M) \tag{3}$$

3) 단어 w가 c번 나타났을 때, 단어 빈도는 c_w로 나타내며, 말뭉치에서 c번 나타난 단어의 개수를 $fof(c)$라고 할 때, 단어 w에 대한 빈도의 빈도는 $f_w = fof(c)$가 된다. 지프의 법칙(Zipf's law)에 의하면 어떤 단어의 빈도(frequency)는 근사적으로 그 단어의 빈도 순위(rank)에 반비례하며[20], 단어 빈도 c와 빈도의 빈도 $fof(c)$에 대하여 $c*fof(c) \simeq (c+1)*fof(c+1)$이 성립함을 의미한다[21].

여기서 M은 N개의 부형태소 타입(type)[4]으로 이루어진 Morfessor 모델로서, 어휘의 최적 분할위치가 표시된 단어목록이다[10]. C는 말뭉치(corpus)로 W개의 어절 토큰(token)[5]으로 구성되고 j-번째 어절은 n_j개의 부형태소 (μ_{jk}) 토큰으로 분할된다. $P(\mu_{jk})$는 해당 부형태소의 빈도와 말뭉치를 구성하는 전체 부형태소의 빈도 비이며, f_{μ_i}는 해당 i-번째 부형태소의 빈도의 빈도를 나타내고, s_{μ_i}는 부형태소 μ_i를 구성하는 문자들의 빈도 열을 나타낸다[10][11]. 분할에 사용되는 전체 비용함수는 수식 (3)의 사후 확률에 음의 로그 함수를 취하여 계산하며, 비용이 최소화되는 곳을 최적의 분할 위치로 표시하여 모델에 저장한다.

Morfessor는 이를 참조하여, 문장 단위의 테스트 말뭉치 (C_{test})에서 띄어쓰기를 제거한 후 수식 (4)와 같이 비터비 (Viterbi) 알고리듬을 이용하여 최대 확률을 갖는 경계(S')를 찾아서 분할한다[11].

$$S' = \arg\max_S P(C_{test}|S,M) \quad (4)$$

2.2.2 Word piece model

어절 단위에서 시작하여 세부적으로 분할하는 Morfessor와 달리, Word piece model (WPM)은 음절 단위에서 시작하여 최대 빈도인 토큰 쌍을 병합하여 확장한다[7]. WPM을 이용한 단위 생성 방법에서는 먼저 어절 단위 말뭉치를 음절 단위로 분할하고, 학습 데이터로부터 비교군(reference group)이 될 초기 언어모델을 생성한 후에, 미리 정한 토큰 개수 또는 우도(likelihood)에 도달할 때까지 빈도가 가장 높은 토큰 쌍을 찾아 순차적으로 병합하는 과정을 반복한다.

3. 제안 방법

제안된 인식단위 결정 방법은 <그림 2>와 같이 형태소 분할, 형태소 내 분할, 분할 단위 간 병합의 3단계로 구성된다.

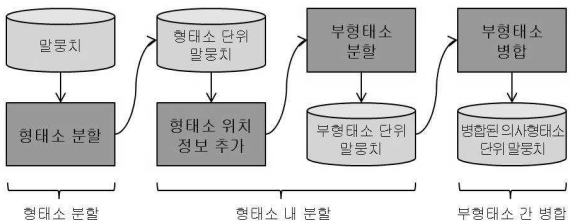

그림 2. 제안 알고리듬 블록도
Figure 2. Block diagram of the proposed algorithm

4) 타입(type)은 토큰(token)을 유형별로 분류한 것으로서, 토큰의 다양성을 나타낸다. 예를 들어 “a b c a b"라는 문장은 5개의 토큰과 3개의 타입으로 이루어져 있다.

5) 토큰의 개수는 중복 발생한 회수를 모두 합산하여 계산된다.

3.1. 형태소 분할

형태소 분할 단계에서는 어절 단위 말뭉치를 형태소 단위로 분할한다. 분할 결과에는 ‘ㄴ’, ‘ㄹ’, ‘ㅁ’, ‘ㅆ’과 같은 단일 자소로 이루어진 형태소가 출력되며, 이러한 단일 자소는 다른 형태소에 비해 길이가 짧아 인식률에 영향을 줄 것이라 예상된다. 하지만, 예비 실험에서 앞서 예시한 4개의 단일 자소들은 다른 자소들에 비하여 발화의 길이가 길어서 단일 자소를 출력하지 않는 의사형태소 단위 말뭉치보다 더 나은 인식결과를 나타내었다. 이러한 이유로 형태소 분할 단계에서는 단일 자소의 출력을 허용하여 분할하였다. <그림 3(a)>는 형태소 분석기에 입력되는 어절단위 말뭉치의 예시이며, <그림 3(b)>는 단일 자소의 출력을 허용한 형태소 단위 말뭉치의 예시이다.

```
길가에 꽃을 심어 가꿉니다
산에 사는 사람들은 나무와 버섯을 가꿉니다
언제 만나도 가족 같습니다
부모님께 카네이션을 달아 드렸다
꿈과 사랑을 그려낸다
사람이 생명을 가지고 있다는 점에서는
(a)
길가 에 꽃 을 심 어 가꾸 ㅂ니다
산 에 사 는 사람 들 은 나무 와 버섯 을 가꾸 ㅂ니다
언제 만나도 가족 같 습니다
부모님 께 카네이션을 달 아 드렸 다
꿈 과 사랑 을 그려 내 ㄴ 다
사람 이 생명 을 가지 고 있다 는 점 에서 는
(b)
```

그림 3. (a) 어절 단위 문장 (b) 형태소 단위 문장
Figure 3. (a) Eojeol-unit sentences (b) morpheme-unit sentences

3.2 형태소 내 분할

3.2.1 형태소 위치 정보 추가

형태소 중에서 “은”, “는”, “이”, “가”와 같은 조사나 “다”, “고”와 같은 어미는 어절의 마지막에 위치하며 다른 형태소에 비해 빈도가 높다. 따라서 빈도를 고려하는 Morfessor와 같은 비교사 분할 방법에서 명사인 “가족” 또는 “-다고 하는”의 준말인 “다는”과 같은 형태소가 빈도가 높은 “가”, “다”에 영향을 받아, “가+족” 또는 “다+는”으로 분할되는 결과를 야기할 수 있다. 본 논문에서는 이러한 문제를 고려하기 위해 <그림 4>과 같이 입력 어절 앞뒤에 밑줄 문자(‘_’)를 붙여 명사 “_가족”의 어절 앞에 나타나는 “_가”와 어절 뒤에 나타나는 조사 “가_”를 구분하였다.

```
_길가 에_ _꽃 을_ _심 어_ _가꾸 ㅂ니다_
_산 에_ _사 는_ _사람 들 은_ _나무 와_ _버섯 을_ _가꾸 ㅂ니다_
_언제_ _만나도_ _가족_ _같 습니다_
_부모님 께_ _카네이션을_ _달 아_ _드렸 다_
_꿈 과_ _사랑 을_ _그려 내 ㄴ 다_
_사람 이_ _생명 을_ _가지 고_ _있다 는_ _점 에서 는_
```

그림 4. 형태소 위치정보가 고려된 말뭉치
Figure 4. Corpus considering the morpheme position

3.2.2 부형태소 분할

부형태소(submorpheme) 분할에 사용되는 Morfessor는 <그림 5>와 같이 어휘의 빈도와 어휘 목록을 이용하여 비용함수가 최소가 되는 분할 경계를 찾아, 이후 다른 말뭉치에서도 적용할 수 있도록 이를 모델로 생성한다. 단계1에서는 입력된 빈도와 단어 목록을 빈도순으로 정렬하고 초기 비용을 계산한다. 이때, <그림 5>에서 1)은 말뭉치 비용함수로 수식 (1)의 결과 값을 가지며, 2)는 어휘 비용함수로 수식 (2)의 M, 3)은 부형태소의 빈도 분포 비용함수로 수식 (2)의 $P(f_{\mu_1},...,f_{\mu_M})$, 4)와 5)는 부형태소를 구성하는 문자들의 빈도 분포 비용함수(수식 (6)의 $P(s_{\mu_i})$)와 문자열의 길이 분포 비용 함수(수식 (6)의 $P(l_{\mu_i})$)로 수식 (2)의 $P(s_{\mu_1},...,s_{\mu_M})$을 구성한다. 이후, 목록에 있는 단어를 선택한 뒤, 분할 가능한 모든 위치에서 두 개의 부형태소로 분리하고, 분리된 두 부형태소를 목록에 추가하여 다시 비용 함수를 계산한다. 단계2에서는 계산된 새로운 비용함수가 기존의 비용함수보다 작거나 미리 설정한 값에 도달할 때 까지 재귀적으로 반복 수행한다. 마지막으로, 어휘의 빈도와 어휘로 구성된 목록을 출력하는데, 이때 분할 경계에는 '+' 기호를 삽입한다.

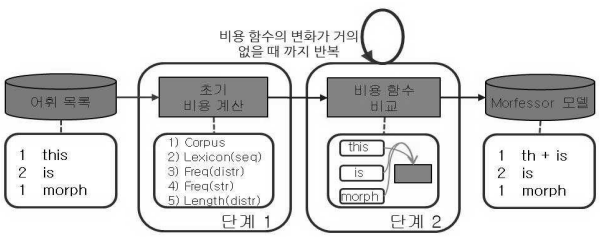

그림 5. Morfessor 모델 생성 블록도
Figure 5. Block diagram of Morfessor model creation

Morfessor Baseline 모델 생성을 위한 입력 단위로 영어의 경우 띄어쓰기 단위인 단어로 이루어진다. 하지만, 한국어의 경우에 영어와 마찬가지로 띄어쓰기 단위인 어절로 어휘 목록을 구성할 경우 <그림 6>과 같이 거의 모든 어절이 분할되지 않는 결과가 나타난다. 이것은 영어의 경우 말뭉치 내에 단어("go"의 3인칭 단수 "goes")의 원형("go")이 존재하지만, 한국어의 경우 대부분의 어절이 여러 형태소의 조합으로 이루어져 있기 때문이다. <그림 6>의 어절 "만나+도"는 예외적인 경우로서 형태소 "만나"가 말뭉치에 독립적으로 발생할 수 있어 분할된 것으로 보인다. 이러한 문제를 고려하여 Morfessor 모델 생성을 위한 입력 어휘 목록은 한국어의 경우 형태소로 이루어지도록 설정하였다.

Morfessor는 모델을 참조하여 띄어쓰기가 제거된 문장 단위의 말뭉치에서 분할한다. 분할 대상이 띄어쓰기가 제거된 문장 단위인 이유는 "우리", "나라"와 같이 자주 연결되어 나오는 단어를 "우리나라"로 합치기 위함이다. Morfessor의 경우 비용함수를 통하여 분할은 가능하지만, 병합의 기능은 존재하지 않기 때문이다. 본 논문에서는 이후 부형태소 간 병합 단계에서 자주 나오는 부형태소를 고려하여 병합할 것이기 때문에 문장 단위가 아닌 어절 단위에서 모델을 참조하여 분할하도록 한다.

길가에 꽃을 심어 가꿉니다
산에 사는 사람들은 나무와 버섯을 가꿉니다
언제 [만나 도] 가족 같습니다
부모님께 카네이션을 달아 드렸다
꿈과 사랑을 그려낸다
사람이 생명을 가지고 있다는 점에서는

그림 6. Morfessor의 어절 단위 분할 결과
Figure 6. Morfessor eojeol-unit segmentation results

형태소 단위로 모델을 생성하였을 경우 앞서 설명한 바와 같이 "ㄴ", "ㅂ니다", "ㅆ다"와 같은 자소가 결합된 형태소가 존재한다. 이러한 상태에서 한글로 구성된 말뭉치를 로마자로 변환하지 않고 그대로 입력으로 넣을 경우 "입니다"가 원하는 결과인 "이+ㅂ니다"가 아닌 "입+니+다"로 출력이 된다. 이는 입력 어절 "입니다"를 자소 단위인 "이+ㅂ니다", "입+니다", "입ㄴ+ㅣ다", "입니+다", "입니ㄷ+ㅏ", "입니다" 중에서 사후확률을 최대로 갖는 것으로 분할하는 것이 아니라, 음절 단위인 "입+니다", "입니+다", "입니다"로 분할하여 사후확률이 최대인 것으로 분할하기 때문이다. 이러한 문제를 해결하기위하여 이전 단계에서 생성한 형태소 단위를 로마자로 바꾼다. 이때, 두 개의 로마자로 표기("eo", "ae")되는 문자의 경우 단일 문자의 로마자("U", "E")로 치환한다. 결과적으로, "입니다"의 경우 "ibnida"로 변환하여, 모델의 "bnida"와 비교 및 분할이 가능하도록 하며, 분할 후 한글로 복원한다.

길 가 에 꽃 을 심 어 가꾸 ㅂ니다
사 ㄴ 에 사 는 사람 들 은 나무 와 버섯 을 가꾸 ㅂ니다
언제 만나 도 가족 같 습니다
부모님 께 카네이션 을 [다 르] 아 드렸 다
꿈 과 사랑 을 그려내 ㄴ다
사람 이 생명 을 가지 고 있 [다 는] 점 에서 는
(a)
길가 에 꽃 을 심 어 가꾸 ㅂ니다
산 에 사 는 사람 들 은 나무 와 버섯 을 가꾸 ㅂ니다
언제 만나 도 가족 같 습니다
부모님 께 카네이션 을 [달] 아 드렸 다
꿈 과 사랑 을 그려내 ㄴ다
사람 이 생명 을 가지 고 있 [다는] 점 에서 는
(b)

그림 7. (a) 형태소 위치를 고려하지 않은 Morfessor의 형태소 단위 분할 결과
(b) 형태소 위치를 고려한 Morfessor의 형태소 단위 분할 결과
Figure 7. (a) Morfessor submorpheme unit without considering morpheme position segmentation
(b) Morfessor submorpheme unit considering morpheme position segmentation

<그림 7(a)>는 형태소 단위를 Morfessor 입력 단위로 사용한 결과이며, <그림 7(b)>는 형태소 위치를 고려한 단위를 Morfessor 입력 단위로 사용한 결과이다. 빈도가 높은 조사인 "가"에 의해 "가족"이 분할될 것이라 예상하였지만, 형태소의 위치정보를 추가하지 않은 경우에도 명사 "가족"은 분할되지 않는다. 이는 명사 "가족"이 말뭉치 내에서 비교적 많이 존재하기 때문으로 보인다. 같은 이유로 "-다고 하는"의 준말인 "다는"의 경우 비교적 낮은 빈도로 존재하기 때문에 형태소 위치를 고려하지 않은 경우 "다+는"으로 분할되지만, 형태소 위치를 고려한 경우 분할되지 않는다.

형태소 단위에서 비교사 분할 방법인 Morfessor을 사용하여 더 세부적으로 분할할 때, 기존의 형태소 단위보다 출력 단위의 길이가 짧아지며 OOV 단어가 감소된다.

수식 (2)는 말뭉치의 어휘(lexicon) 비용함수로써 빈도 분포 비용함수 $P(f_{\mu_1},...,f_{\mu_M})$와 길이 분포 비용함수$P(s_{\mu_1},...,s_{\mu_M})$으로 구성되어 있다. 빈도 비용함수의 경우 부형태소의 사용 빈도에 대한 확률 분포인 빈도의 빈도 분포 $P(f_{\mu_i})$를 나타내며, 이는 균일 분포와 지프의 법칙(Zipf's law) [9] 중에서 원하는 분포를 설정하여 비용함수를 구한다.

지프의 법칙을 사용할 경우, 빈도 1인 부형태소 타입 개수와 전체 부형태소 타입 개수의 비율인 hapax legomenon (h)[9]를 이용한다. 수식 (5)와 (6)은 지프의 법칙을 나타낸다 [9].

$$P(f_{\mu_1},...,f_{\mu_M}) = \prod_{i=1}^{M} P(f_{\mu_i}) \tag{5}$$

$$P(f_{\mu_i}) = f_{\mu_i}^{\log_2(1-h)} - (f_{\mu_i}+1)^{\log_2(1-h)} \tag{6}$$

길이 분포 비용함수의 경우, 부형태소를 구성하는 각각의 문자(character)에 대한 빈도 분포 $P(s_{\mu_i})$와 문자의 길이에 대한 확률 분포 $P(l_{\mu_i})$로 구성되어 있다.

$$P(s_{\mu_1},...,s_{\mu_M}) = \prod_{i=1}^{M} P(s_{\mu_i})P(l_{\mu_i}) \tag{7}$$

이때, i-번째 부형태소 μ_i의 문자열 길이(length; l_{μ_i})에 대한 확률 분포 $P(l_{\mu_i})$는 지수 확률 분포와 감마 확률 분포 중에서 선택하여 비용함수를 구할 수 있다[9].

감마 함수와 감마 확률 분포는 수식 (8), (9)와 같이 나타난다.

$$P(l_{\mu_i}) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} l_{\mu_i}^{\alpha-1} e^{-l_{\mu_i}/\beta} \tag{8}$$

$$\Gamma(\alpha) = \int_0^{\infty} z^{\alpha-1} e^{-z} dz \tag{9}$$

적절한 파라미터 α와 β를 이용하여 평균값($\alpha\beta$)을 설정할 수 있다.

본 논문에서는 이전 단계에서의 OOV 단어 감소를 극대화하기 위해 기존의 길이 분포 비용함수에 수식 (10)과 같이 가중치(λ)를 적용하여 기여도를 높여, 기존의 Morfessor 분할 결과보다 더욱더 세분화된 분할 결과를 유도한다.

$$\log P(s_{\mu_1},...,s_{\mu_M}) = \sum_{i=1}^{M} \log P(s_{\mu_i}) + \lambda \sum_{i=1}^{M} \log P(l_{\mu_i}) \tag{10}$$

3.3. 부형태소 간 병합

기존의 형태소는 앞의 분할 단계에서 더 작은 단위로 분할된다. 그 결과, OOV 단어는 줄어들지만 평균 발화길이 또한 줄어들어 인식 결과에 영향을 미치게 된다. 이번 단계에서는 이러한 문제점을 보완하기 위하여 발생 빈도가 높은 부형태소 쌍을 병합한다.

병합 방법으로는 형태소 위치정보가 고려되며, 부형태소를 한글로 변환하기 전, 로마자로 이루어진 말뭉치를 어절 내에서 두 개씩 연결하여 목록을 생성한다(그림8(a)). 이후, 생성된 목록에서 빈도가 높은 부형태소 쌍 1,000개, 2,000개, 3,000개를 후보로 하여 말뭉치에 적용한다(그림8(b)). 실제 실험에서는 한글을 로마자로 변경 후, 'ㄲ', 'ㅊ', 'ㅘ'와 같은 이중 모음과 겹자음을 'G', 'C', 'W'와 같은 독립된 단일 문자의 로마자로 치환하여 자소 간의 병합이 이루어지도록 한다.

길가에	346855 한_
_꽃을_	186927 인_
_심어_	179764 이다_
가꿉니다	157498 하는_
_산에_	140412 있다_
_사는_	134055 _있는_
_사람들	133416 할_
들은_	115922 들이_
나무와	113568 _것이
버섯을	109554 입니다_
가꿉니다	94956 들은_
(a)	(b)

그림 8. (a) 어절 내에서 두 개씩 연결한 쌍의 목록
(b) 빈도순으로 정렬한 연결 쌍의 목록
Figure 8. (a) List of concatenated pairs within an eojeol
(b) List of concatenated pairs sorted in the non-decreasing order

길가 에 꽃 을 심 어 가꾸 ㅂ니다
산 에 [사는] 사람 [들은] 나무 와 버섯 을 가꾸 ㅂ니다
언제 만나 도 가족 같 습니다
부모님 께 카네이션 을 달 아 드렸 다
꿈 과 사랑 을 그려내 ㄴ다
[사람이] 생명 을 [가지고] 있 다는 점 [에서는]

그림 9. 1,000개의 부형태소를 병합한 결과
Figure 9. Results after merging 1,000 submorphemes

실험결과를 보면, 가장 빈도가 높은 병합 대상은 "하+ㄴ"이며, 다음으로 "이+ㄴ", "이+다"로 나타났다. 빈도가 높은 부형

태소 쌍 1,000개를 예제 말뭉치에 적용한 결과 <그림 9>와 같은 결과가 나타난다. 이전 단계의 실험결과인 <그림 7>의 (b)와 달리 “들+은”, “가지+고” 등이 병합된 것을 확인할 수 있다.

한편, WPM에 의한 분할 방법에서는 <그림 10>과 같이 음절 단위에서 빈도를 고려하여 토큰 쌍이 연결(concatenation)된다[7]. 이 방법은 어근과 어미의 연결로 이루어진 한국어에 적용할 경우 자소를 고려할 수 없다는 문제점이 발생한다. 이에 반하여, 본 논문에서 제안한 방법에서와 같이 개선된 비용함수를 사용하여 생성된 부형태소 단위에서 병합할 경우에는 자소를 고려하여 병합할 수 있게 되어 더 나은 인식결과를 나타낼 것으로 보인다.

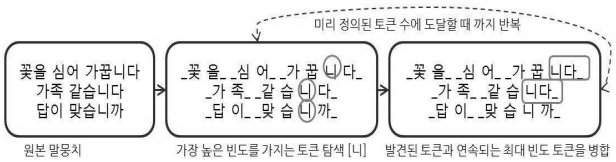

그림 10. WPM 기반 병합 예제
Figure 10. Example of WPM-based unit merging

4. 인식단위 실험 결과

4.1. 말뭉치

언어모델 생성에 사용된 3억 어절의 한글 말뭉치는 초등학교 교과서 51만 어절, 고등학교 교과서 7만 어절, 문학 300만 어절, 비문학 350만 어절, 국어정보베이스 말뭉치 1,000만 어절, 방송뉴스 1,500만 어절, 날씨, 시사, 경제 등 여러 가지 분야로 구성된 1990년도에서 1999년 사이에 발간된 신문기사 2.6억 어절을 사용하였다. 실험에 불필요한 문장 기호, 특수문자는 제거하였으며, 영어의 경우 한국어 발음으로 변경 후 사용하였다.

4.2. 빈도 및 길이 비용함수

Morfessor 모델 생성을 위한 입력단위는 어절단위에서 형태소 단위로 변경하여 실험하였다. 형태소 분석을 위하여 한국어 음성인식 플랫폼 Echos [12]에 포함된 형태소 분석기를 사용하였으며, 이는 약 10만 형태소 사전을 가진다.

한국어의 경우 Morfessor 모델을 참조하여 말뭉치를 분할할 때, 문장 단위의 말뭉치에서 모델을 통해 분할한 결과보다는 어절 단위로 변경된 말뭉치에서 분할한 결과에서 더 빠른 분할이 이루어졌으며, 어절 범위를 넘어선 병합이 일어나지 않아 더 나은 인식결과를 얻을 수 있었다.

빈도 비용함수로 수식 (5), (6)에 나타난 지프의 법칙을 사용하였다. 말뭉치에서 hapax legomenon을 구하여 지프의 법칙 공식에 적용한 결과, 실제 말뭉치의 빈도 확률과 비슷한 분포를 가지는 것을 확인하였으며, <그림 11>은 빈도 1에서 10까지의 형태소 빈도의 빈도(fof) 분포이다. 여기서 사용된 hapax legomenon는 빈도 1인 형태소 타입 개수 111만 개와 전체 형태소 타입 개수 270만 개의 비(rate)인 $h=0.41$의 값을 사용하였다.

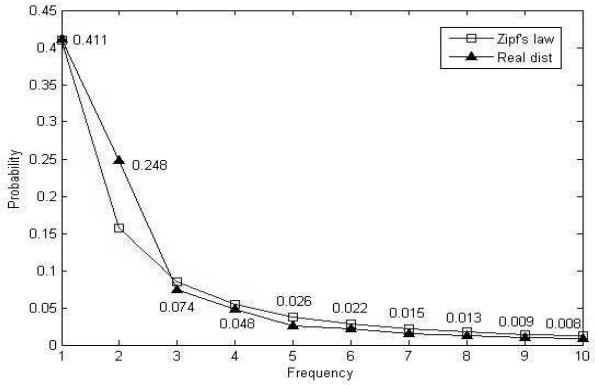

그림 11. 지프의 법칙 및 실제 형태소 빈도의 빈도 확률 분포
Figure 11. Zipf's law and real probability distribution of frequency-of-frequency of morpheme

길이 비용함수로 수식 (8), (9)의 감마 확률분포를 사용하였다. 형태소 단위로 분할 후 로마자로 변경한 상태에서의 전체 길이 빈도는 6억 개로, 각각의 길이 빈도를 통해 형태소 자소 길이의 평균값이 4.1임을 구할 수 있었다. <그림 12>는 α와 β를 각각 4.4, 0.9로 설정할 때의 형태소 길이 감마 분포이다. 감마 확률 분포 실험을 통해 확인한 결과 수식 (2)에 구성된 비용함수들이 수식 (1)의 비용함수에 비해 기여도가 낮았다.

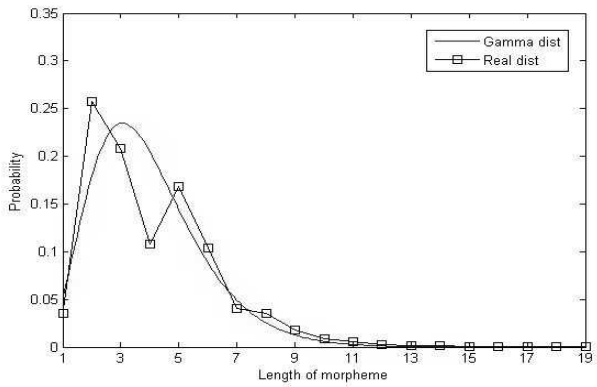

그림 12. Gamma 분포와 실제 말뭉치의 형태소 길이 분포
Figure 12. Gamma distribution and real distribution of morpheme length

<표 1>은 형태소 단위 말뭉치 그리고 길이분포 비용함수의 가중치를 변경시켜가면서 분할한 말뭉치의 부형태소 타입 개수이다. 어절 단위 말뭉치의 경우 전체 1,115만개의 어절 타입을 가지고 있으며, 형태소 단위 말뭉치의 경우 270만개의 형태소 타입 개수를 가지고 있었다. 이때, 생성된 형태소 타입 목록에서 빈도 2 이상의 형태소 타입 개수는 159만개로, 나머

지 111만개의 경우 띄어쓰기 오류, 외래어, 오탈자, 형태소 분석기의 성능에 따라 처리되지 않은 고유명사 등으로 구성된 것으로 보인다. 또한, 가중치가 증가할수록 부형태소 타입 개수가 줄어드는 것으로 보아 더 작은 단위로 분할되고 있는 것을 확인하였다.

표 1. 가중치 변화에 따른 말뭉치 부형태소 개수
Table 1. Number of submorpheme entries with a varying weight

λ	10	30	50	70	100
부형태소개수	420k	274k	223k	196k	172k

4.3. 부형태소 병합 개수

부형태소 쌍의 후보 개수에 따르는 말뭉치 평균 길이 분포를 확인해 보면 <표 2>와 같다. 병합을 하지 않을 경우 말뭉치 내 평균 음절 길이는 2.48로, 병합 대상이 많아질수록 말뭉치 내 평균 음절 길이는 증가하며, 병합 대상이 많아질수록 증가 속도는 점차 감소한다.

표 2. 병합 부형태소 개수 별 평균 음절 길이 분포
Table 2. Average syllable length for different numbers of merged submorpheme

병합개수	0	1,000	2,000	3,000
평균음절길이	2.48	2.94	3.03	3.09

5. 음성인식 실험 결과

5.1. 음성 데이터베이스

한국어 연속 음성인식 데이터베이스로는 ETRI에서 개발한 음운균형문장(phonetically balanced sentence; PBS) 데이터베이스를 사용하였다. 이 데이터베이스는 음성인식 및 합성 등 우리말 음성정보처리시스템의 개발을 위한 다양한 음소 환경이 포함된 대어휘 낭독체 음성으로 구성되어 있으며, 조용한 사무실 환경에서 녹음되었으며, 16 kHz, 16 bit PCM으로 샘플링 되었다. 실험을 위해 텍스트 독립 및 화자 독립이 되도록, 80명의 화자가 8,361개의 문장을 총 18시간 동안 발화한 학습데이터와 20명의 화자가 6,414개의 어절로 이루어진 500개의 문장을 1.5시간 동안 발화한 테스트 데이터로 나누어 실험하였다.

5.2. 실험 환경

음성인식기에 사용된 특징벡터로는 1차 미분과 2차 미분이 포함된 39차 mel-frequency cepstral coefficients (MFCC)를 사용하였다. 음향모델로는 3개의 상태(state)로 구성된 HMM을 사용하였으며, 기본 음소 개수는 40개이다. 음향모델을 학습하기 위하여 Kaldi Script 중에서 WSJ/s1 [15]을 사용하였다. 모노폰 학습 단계에서는 가우시안 혼합 모델의 초기 가우시안 개수 300개에서 시작하여 1,000개가 될 때까지 가우시안 분포를 분할하였으며, 트라이폰 학습 단계에서는 모노폰 모델로부터 시작하여 전체 가우시안 개수가 10,000개가 되도록 분할하였고, 트라이폰 문맥을 고려하기 위하여 2,000개의 잎사귀 노드(leaf node)로 구성된 최적화된 결정트리(decision tree)를 사용하였다.

언어모델을 구하기 위하여 SRILM Toolkit [13]을 사용하였으며, 10만 인식단위를 갖는 트라이그램을 구하였다. 트라이그램 언어모델을 적절한 크기 이하로 유지하기 위하여 바이그램과 트라이그램에 cutoff 3을 적용하였다.

인식 과정에서 발생하는 메모리 부족 문제를 해결하기 위하여, 디코더로는 Kaldi BigLmDecoder [14]를 사용하였으며, 빔 크기(beam size)는 13으로 설정했다.

또한, 음절 단위 오류율을 구하기 위하여, 의사형태소 단위로 출력된 인식 결과를 자소가 존재하지 않는 한글 문장으로 변경 후, 다시 음절 단위로 분리하여 참조(reference) 음절열과의 정렬을 통하여 얻은 삽입/탈락/대치 오류를 합산하였다.

5.3. 실험 결과

제안된 인식단위의 성능평가를 위해 복잡도(perplexity; PP), 어휘외 단어율(OOV rate), 음절 오류율(syllable error rate; SER)을 비교하였다. 이때, 복잡도(PP)는 테스트 말뭉치에 대해 구하였으며, 다음 값으로 계산된다[13].

$$PP = 10^{-\frac{\log prob}{N_{sen}+N_{word}}} \tag{11}$$

$\log prob$는 언어모델(M)이 주어질 때 테스트 말뭉치(C)의 로그확률(log-probability)인 $\log P(C|M)$을 나타내며, N_{sen}은 테스트 말뭉치의 문장 개수, N_{word}는 인식단위(어절/의사형태소/부형태소)의 개수를 나타낸다.

5.3.1. 어절 단위 인식 (실험 I)

어절 단위 인식 실험의 경우 10만 어절 단위 인식 결과는 <표 3>에 정리된 바와 같이 1,925의 복잡도와, 25.5%의 높은 어휘외 단어율을 나타내었다. 이를 줄이기 위해 말뭉치 내 모든 어절을 인식 어휘에 포함하도록 변경하여, 1,200만 어절 단위 인식 결과 어휘외 단어율은 3.8%로 줄어들었지만, 복잡도는 13,825로 크게 증가하였다. 이와 같은 결과로 인하여 음절 오류율이 높게 나타날 것이 예측 가능하며, 1,200만 어절 단위를 사용하는 경우 메모리 부족으로 음성인식기를 실행할 수 없었기 때문에 더 이상 실험하지 않았다.

5.3.2. Morfessor 분할 단위 인식 (실험 II)

Morfessor 모델 생성을 위한 입력 단위는 영어나 핀란드어의 경우 단어의 빈도와 단어의 목록으로 입력된다. 같은 방법으로 어절 단위를 Morfessor에 입력할 경우, 10만 어절 단위 인식 결과에 비해 어휘외 단어율은 18.3%로 줄어들지만, 복잡도는 3,039로 높아졌다(<표 3> 실험 II 결과). 또한, 소량의 말뭉치와 비교하여 확인해본 결과, 말뭉치의 크기가 늘어날수록 어휘외 단어율 감소 효과는 줄어들었으며, 수식 (1)의 말뭉치 비용함수 $P(C|M)$의 증가량에 비하여 수식 (2)의 모델 비용함수 $P(M)$의 증가량이 비교적 적게 나타났다. 실험 I에서와 같은 이유로 음성인식 실험은 하지 않았다.

5.3.3. 형태소 단위 인식 (실험 III)

형태소 단위 인식기는 본 논문의 인식단위 성능평가를 위한 베이스라인이 된다. 언어학적 정보를 사용하는 형태소 단위 인식 실험 결과, 139의 낮은 복잡도를 나타냈으며, 어휘외 단어율 또한 2.1%으로 기존의 어절단위에 비하여 감소하였고, 22.1%의 음절 오류율을 가진다(<표 3> 실험 III 결과). 발생한 어휘외 단어는 “디에이치엘”, “팜프렛”, “파운드”와 같은 외래어나, “도섭”, “윤처사”, “마영감”과 같은 형태소 사전에 존재하지 않는 고유명사였다.

5.3.4. 부형태소 단위 인식 (실험 IV, 실험 V)

Morfessor 모델 생성을 위한 입력 단위로 형태소 단위를 입력할 경우, 분할된 말뭉치에서 ‘ㄴ’, ‘ㄹ’, ‘ㅁ’, ‘ㅆ’과 같은 단일 자소들이 많이 나타났으며, 빈도가 높은 조사들에 의한 무분별한 분할로 인식단위의 평균 지속시간이 매우 짧아졌다. 결과적으로 어휘외 단어율이 1.2% 감소하였으며, 음절 오류율은 21.4%로 감소하였다(<표 3> 실험 IV 결과).

형태소 위치정보를 추가한 단계에서는 21.0%의 음절 오류율이 나타났다(<표 3> 실험 V 결과). 이는 위치정보를 적용하지 않았을 때 발생하던 단일 자소들이 형태소 위치정보를 추가함으로써 줄어들었으며, 빈도가 높은 조사에 대한 무분별한 분할이 억제되었기 때문으로, 인식 단위의 평균 지속시간은 늘어나고 어휘외 단어율도 형태소 단위에 비해 줄어 음절 오류율이 감소하였다.

Morfessor의 비용함수에 가중치를 적용한 결과 형태소의 위치정보를 고려하며, 인식 단위를 더 세분화 되게 분할할 수 있었다. 그 결과, 복잡도가 낮아지고, 어휘외 단어율이 줄어들며, 음절 단위 오류율이 감소하였다. 계속해서 가중치를 증가시킨 경우에는, 복잡도가 줄어들었지만, 길이 비용함수 외에 다른 비용함수들이 무시되어 부형태소의 길이에 의존한 분할이 이루어졌으며, 음절 오류율이 증가하였다.

<표 4>에서와 같이 길이 분포 비용함수에 $\lambda = 50$을 적용한 결과, 음절 오류율이 상대적으로 8.6% (=(22.1 − 20.2) / 22.1 ×100)감소되었다.

표 3. 단계별 실험의 음성인식 결과
Table 3. Speech recognition results in the step-by-step experiments

	PP	OOV rate (%)	SER (%)
실험 I	1,925	25.5	-
실험 II	3,039	18.3	-
실험 III	139	2.1	22.1
실험 IV	158	0.9	21.4
실험 V	162	1.2	21.0

표 4. 가중치 변화에 따른 인식 결과
Table 4. Recognition results with a varying weight λ

λ	10	30	50	70	100
PP	162	164	160	155	154
OOV rate (%)	0.9	0.6	0.3	0.1	0.0
SER (%)	20.7	20.4	20.2	20.3	20.4

5.3.5. 병합된 의사형태소 단위 인식

부형태소 분할 실험에서 가장 낮은 음절 오류율을 가지는 분할 단계에서 어절 범위 내에서 두 개씩 조합하였을 때 높은 빈도순으로 병합 실험하였다. 복잡도를 구하기 위해 기존의 수식 (11)를 그대로 사용하였을 때, 병합 개수에 따라 로그확률(log-probability)은 줄어들었지만, 분모에 위치한 토큰 개수 또한 줄어들어 전체 복잡도가 증가되는 것을 확인할 수 있었다. 이러한 문제를 해결하기 위해, 이번 실험에서는 수식 (11)의 분모에 있는 토큰 개수(N_{word})를 병합 전 토큰 개수(C_{word}; 본 논문에서는 12,361)와 동일하게 고정시켜 비교하였다.

$$PP_{new} = 10^{-\frac{\log prob}{N_{sen}+C_{word}}} \quad (12)$$

부형태소의 병합에 따른 인식 결과에서는 단위의 발화길이가 길어짐에 따라 음절 오류율이 낮아졌으며 병합되는 개수가 늘어날수록 점차 음절 오류율이 높아지는 것으로 나타난다. 이는 기존의 10만 어휘 인식에 병합된 부형태소들이 추가되어 인식 어휘수가 늘어났기 때문인 것으로 보인다.

병합 개수로 1,000개를 적용한 결과 19.0%의 음절 오류율을 나타냈으며, 이는 14.0%의 상대적 음절 오류율 감소를 의미한다. 또한, 새로 계산한 복잡도(PP_{new})는 음절 오류율과 유사한 변화를 보인다는 것을 확인할 수 있어서, 음성인식 성능을 예측할 수 있는 효과적인 지수라고 판단된다.

표 5. 병합 수에 따른 인식 결과
Table 5. Recognition results with the varying number of merged morphemes

병합개수	0	1,000	2,000	3,000
PP	160	316	354	386
$\log prob$	-28240	-28033	-28051	-28118
PP_{new}	160	151	152	154
#Unit	12,361	10,746	10,541	10,405
SER (%)	20.2	19.0	19.1	19.2

5.3.6. 무역 상담 데이터베이스 인식 실험

제안된 알고리듬의 성능을 기존 연구 결과와 비교하기 위하여 추가적으로 무역상담 관련 한국어 연속 음성 데이터베이스[16]를 사용하여 제안한 인식단위의 성능을 조사하였다. 실험 환경으로는 조용한 사무실 환경에서 녹음되었으며, 샘플링은 16 kHz, 16 bit PCM 형식으로 이루어졌다. 이 데이터베이스는 시간, 날짜, 지역명 등과 무역에 관련된 단어들을 포함하는 낭독체 음성이다. 인식 성능 평가를 위해 30명의 화자가 개의 18,879개의 어절로 이루어진 2,965개의 문장을 약 2.3시간 동안 발화한 테스트 데이터를 사용하였다.

제안된 인식단위의 성능 평가를 위해 테스트 데이터만을 기존의 음운균형문장 데이터베이스에서 무역상담 관련 데이터베이스로 변경하여 음성인식 성능을 비교하였다. 형태소 단위에서 제안된 인식 단위로 변경한 결과 1.1%의 어휘외 단어율이 감소하였으며, 9.7%의 상대적 오류율이 감소되었다.

표 6. 무역상담 음성인식 결과
Table 6. Trade-related speech recognition results

	형태소 단위	제안된 단위
PP	140	256
OOV rate (%)	1.3	0.2
SER (%)	24.8	22.4

이전 연구 결과[17]에서는 본 논문과 동일한 무역상담 데이터 중에서 8명의 화자로 구성된 799개의 문장을 인식 실험한 경우에 트라이폰 단위에서 10.5%의 단어 오류율을 나타내었다. 이 결과는 학습 말뭉치에 테스트 데이터와 동일한 말뭉치를 포함하는 닫힌 어휘(closed vocabulary) [18]의 경우이기 때문에 높은 성능을 낼 수 있었으며, 본 논문의 결과는 학습 말뭉치와 테스트 데이터가 독립인 열린 어휘(open vocabulary) [18]이기 때문에 낮은 성능을 나타내었다.

5.3.7. 구글 음성인식기와의 성능 비교

구글에서는 음성인식 성능 평가를 위해 Speech API [19]를 제공한다. 위 API를 이용하여 두 테스트 데이터의 성능을 비교한 결과 음운균형문장에서 17.1%, 무역상담 관련 문장에서 14.9%의 음절 오류율이 나타나며, 본 논문에서 제안된 알고리듬의 음절 오류율은 음운균형문장에서 19.0%, 무역상담 관련 문장에서 22.4%로 구글 음성인식기보다 오류율이 높게 나타났다.

표 7. 구글 음성인식기와 제안된 알고리듬의 음절 오류율 비교
Table 7. Comparison of SER (%) using the Google speech recognizer and the proposed algorithm

데이터베이스	제안된 알고리듬	구글 음성인식기
PBS	19.0	17.1
무역상담	22.4	14.9

이러한 차이는 언어모델 및 음향 모델의 차이와, 출력 결과를 정상적인 문장으로 다듬어 주는 후처리에 의한 차이로 보인다. 특히 언어모델은 훈련에 사용한 말뭉치의 크기, 문장 스타일, 토픽, 장르에 매우 민감한데, 본 논문에서 사용한 한국어 말뭉치는 신문기사 내용이 대다수를 차지하는 것에 반해, 구글 음성인식기의 경우 대용량 언어모델을 사용하는 것으로 보인다.

5.3.8. 다중발음을 고려한 언어모델

음성인식 성능을 더욱 향상시키기 위하여 다중발음을 고려한 언어모델[1][3]을 적용하였다. 다중발음을 고려한 발음사전과 언어모델을 생성하기 위하여 부형태소 단위에서 구한 자소열(graphemes; 그림 13(a))과 어절 단위에서 구한 음소열(phonemes; 그림 13(b))을 동적 시간 정합(dynamic time warping; DTW) 알고리듬을 이용하여 정렬한 다음, 의사형태소 단위에 DTW로 정렬된 발음열을 덧붙인(그림 13(c)) 말뭉치를 사용하였다. “다[tt-a]”, “의[i]”와 같은 2만개의 다중발음을 발음사전과 언어모델에 추가한 결과, 구글과 동일한 17.1%의 음절 오류율을 달성하였다.

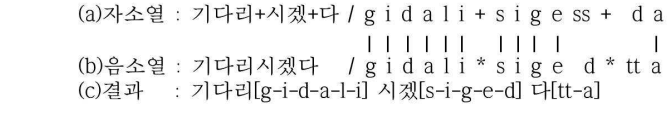

그림 13. 부형태소 기반의 다중발음 생성 예제
Figure 13. Example of building submorpheme-based multiple pronunciations

6. 결론

본 논문에서는 대어휘 연속 음성인식 단위를 정하기 위하여, 의사형태소 분할 결과에 형태소 위치정보와 길이 분포 가중치를 적용하여 비교사 방식으로 분할한 다음 부형태소 간 병합을 적용하는 새로운 방법을 제안하였다. 음운균형문장 데이터베이스를 이용하여 음성인식 실험을 수행한 결과, 제안한 방법은 1.8%의 OOV 감소와, 19.0%의 음절 오류율을 보였으며, 14.0%의 상대적 음절 오류율이 감소되었다. 다중발음을 고려한 언어모델을 추가적으로 적용함으로써 17.1%의 음절 오류율을 나타내는 연속음성인식기를 얻을 수 있었다.

본 논문은 기존의 형태소 분석기로 해결하기 어려운 고유명사나 신조어, 외래어, 복합어와 같은 미등록어를 비교사 방법으로 분할함으로써 어휘외 단어율을 줄이고, 빈도가 높은 단어를 병합함으로써 인식 성능을 향상시키는 효과를 가진다.

참고문헌

[1] Kwon, O.-W., Hwang, K. & Park, J. (1999). Korean large vocabulary continuous speech recognition using pseudomorpheme units. *Proc. EUROSPEECH,* 483-486.

[2] Yu, H.-J., Kim, H., Choi, J.-S. & Hong, J.-M. (1998). Automatic recognition of Korean broadcast news speech. *Proc. ICSLP.*

[3] Kwon, O.-W. & Park, J. (2003). Korean large vocabulary continuous speech recognition with morpheme-based recognition units. *Speech Communication,* Vol. 39, No. 3-4, 287-300.

[4] Creutz, M. & Lagus, K. (2002). Unsupervised discovery of Morphemes. *Proc. ACL-02 Workshop on Morphological and Phonological Learning,* 21-30.

[5] 김영택, 옥철영, 이호석, 윤덕호, 강승식, 심광섭, 윤성희, 서병락, 이재원, 김유섭, 이종우, 오장민, 김선, 권혁철, 서영훈, 이근배, 문유진, 이하규, 장병탁, 양재형, 양승현, 김성동, 박성배, 장정호, 황규백, 신형주. (2001). *자연언어처리.* 서울 : 생능출판사.

[6] Creutz, M. (2006). *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition,* Ph.D. Dissertation, Helsinki University of Technology, Finland.

[7] Schuster, M. & Nakajima, K. (2012). Japanese and Korean voice search. *Proc. ICASSP,* 5149-5152.

[8] Creutz, M. & Lagus, K. (2006). Morfessor in the Morpho Challenge. *Proc. PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*.

[9] Creutz, M. (2003). Unsupervised segmentation of words using prior distributions of morph length and frequency. *Proc. ACL-03,* 280-287.

[10] Siivola, V., Hirsimaki, T., Creutz, M. & Kurimo, M. (2003). Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. *Proc. EUROSPEECH*, 2293-2296.

[11] Hirsimaki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S. & Janne. (2006). Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech & Language,* Vol. 20, No. 4, 515-541.

[12] Kwon, O.-W., Kim, H., Kwon, S., Yun, S., Jang, G., Kim, Y.-R., Kim, B.-W., Yoo, C., & Lee, Y.-J. (2007). Development of a Korean large vocabulary continuous speech recognition platform (ECHOS). *Proc. O-COCOSDA,* 108-111.

[13] Stolcke, A. (2002). SRILM - An extensible language modeling toolkit. *Proc. INTERSPEECH,* 901-904.

[14] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., & Vesely, K. (2011). The Kaldi speech recognition toolkit. *Proc. ASRU,* 1-4.

[15] Downloading Kaldi. http://kaldi.sourceforge.net/install.html.

[16] 박종렬, 권오욱, 김도영, 최인정, 정호영, 은종관. (1995). 한국어 음성 인식을 위한 음성 데이터 수집. 음향학회지, 14권 4호, 74-81.

[17] 최인정, 권오욱, 박종렬, 박용규, 김도영, 정호영, 은종관. (1995). 대용량 한국어 연속음성인식 시스템 개발. 음향학회지, 14권 5호, 44-50.

[18] Jurafsky, D. and Martin, J. H. (2008). Speech and Language Processing, 2e. 95.

[19] Openmoko wiki. (2012). Google Voice Recognition. http://wiki.openmoko.org/wiki/Google_Voice_Recognition.

[20] Zipf's law, http://en.wikipedia.org/wiki/Zipf%27s_law.

[21] Jurafsky, D. and Martin, J. H. (2008). Speech and Language Processing, 2e. 4.5.2 Good-Turing Discounting.

• **방정욱 (Bang, Jeong-Uk)**
충북대학교 제어로봇공학전공
충북 청주시 서원구 내수동로 52(개신동)
Email: isaac@cbnu.ac.kr
관심분야: 음성인식, 음성 및 오디오 처리
현재 제어로봇공학과 석사과정 재학 중

• **권오욱 (Kwon, Oh-Wook)** 교신저자
충북대학교 전자공학부
충북 청주시 서원구 내수동로 52(개신동)
Tel: 043-261-3374
Email: owkwon@cbnu.ac.kr
관심분야: 음성인식, 감정인식, 음성신호처리
2003~현재 충북대학교 전자공학부 교수