언어모델 군집화와 동적 언어모델 보간을 통한 음성인식 성능 향상

정의석°, 전형배, 정호영, 박전규
음성처리연구실, 한국전자통신연구소
.eschung@etri.re.kr, hbjeon@etri.re.kr, hjung@etri.re.kr, jgp@etri.re.kr

LM Clustering based Dynamic LM Interpolation for ASR N-best Rescoring

.Euisok Chung°, Hyung-Bae Jeon, Ho-Young Jung, Jeon-Gue Park
Spoken Langauge Processing Research Section, ETRI

요 약

일반영역 음성인식은 n-gram 희소성 문제로 인해 대용량의 언어모델이 필요하다. 대용량 언어모델은 분산형 모델로 구현될 수 있고, 사용자 입력에 대한 동적 언어모델 보간 기술을 통해 음성인식 성능을 개선할 수 있다. 본 논문은 동적 언어모델 보간 기술에 대한 새로운 접근방법을 시도한다. 텍스트 군집화를 통해 주제별 언어모델을 생성한다. 여기서 주제는 사용자 입력 영역에 대응한다. 본 논문은 사용자 입력에 대하여 실시간으로 주제별 언어모델의 보간 가중치 값을 계산하는 접근 방법을 제시한다. 또한 언어모델의 보간 가중치 값 계산의 부담을 감소하기 위해 언어모델 군집화를 통해 대용량 언어모델 보간 접근 방법의 연산 부담을 해소하기 위한 시도를 한다. 주제별 언어모델에 기반하고 언어모델 군집화를 통한 동적 언어모델 보간 기술의 실험 결과 음성인식 오류 감소율 6.89%를 달성했다. 또한 언어모델 군집화 기술은 음성인식 정확도를 0.09% 저하시켰을 때 실행 시간을 17.6% 개선시키는 실험결과를 보였다.

주제어: 언어모델 동적 보간 기술, 텍스트 군집화, 언어모델 군집화, 음성인식 리스코링

1. 서론

일반영역 음성인식 기술은 이미 상용화에 성공한 기술이라 볼 수 있다. 그러나 모든 영역의 어휘들을 인식할 수 있어야 하고, 해당 어휘 수는 문장 형태의 음성인식을 위해 필요한 n-gram의 수를 가중시킨다. 최근 활발히 연구되고 있는 심층 신경망 (deep neural network) 기술은 n-gram 희소성 문제에 대한 대안이 될 수 있으나, 일반 영역 어휘수를 학습하기에는 아직 미흡하다는 판단이다. 따라서 본 논문은 전통적인 n-gram 언어모델에 기반을 둔 분산 언어모델링 기술로 일반영역 음성인식 성능 향상에 접근한다.

분산 언어모델은 분산된 서버에서 서비스 되는 분산 n-gram 빈도수 서버형태로 처음 제시되었다[1]. 여기서 학습 코퍼스를 서픽스 배열(suffix array) 기술을 이용하여 분리하고, 클라이언트/서버 프레임워크로 접근하였다. 다른 접근 방법으로 MapReduce 프로그래밍 기술을 이용하여 대용량 언어 리소스를 구축하고, 필요에 따라 1-pass 언어모델을 생성하고, 후처리 방식으로 분산 언어 모델을 따로 구축하여 음성인식 성능을 향상 시킨 결과도 보고 되었다[2]. 또한 [3]은 트라이 DB 기반 언어모델 서버와 분산 언어모델의 보간 가중치를 최적화하는 접근 방법을 제시하였는데, 해당 최적화 값 계산에 음성인식 성능 값을 이용하여 학습 비용을 가중시키는 문제가 있었다.

실시간 사용자 입력에 대응하여 동적으로 언어모델들의 보간 가중치를 결정하여 사용자 입력에 적응하는 연구는 [4]에 기술되어 있다. 여기서는 미리 정의된 보간 가중치가 모바일 사용자 입력의 주제 영역에 따라 적용되는 접근 방법을 제안하였다. 문제점은 사용자 입력의 주제 영역이 고정되어 있어야 한다는 점이다. 즉, 새로운 주제 영역이 생성되었을 경우 주제별 보간 가중치가 다시 학습되어야 한다.

본 논문은 일반 영역 대상 동적 언어모델 보간 기술을 제안한다. 여기서 언어모델의 주제별 구성은 텍스트 군집화 기술을 이용하고, 동적 언어모델 보간 가중치 계산은 주제별 언어모델의 입력 음성인식 결과에 대한 언어모델 값을 그대로 이용하는 접근 방법을 취한다. 또한, 주제별 언어모델들을 군집화 하여 연산량 개선을 시도한다.

2. 주제별 언어모델 생성과 동적 언어모델 보간

언어모델을 생성하기 위한 텍스트 코퍼스는 가용 가능한 다양한 영역에서 수집된다. 수집된 텍스트 코퍼스가 주제별로 분리될 수 있다면 주제별 언어모델 생성이 가능하게 된다. 본 논문은 텍스트 군집화 기술을 이용하여 주제별 언어모델을 구축한다. 텍스트 군집화 기술은 주제별 언어 모델 생성 뿐 아니라, 다수 구축되는 주제별 언어모델들을 군집화 하는 단계에서도 활용된다.

텍스트 군집화는 큰 텍스트를 작은 텍스트들로 분리하는 기술이다. 일반적으로 K-means 알고리즘을 이용할 수

있고, 대용량 텍스트의 경우 효율적 접근 방법을 위한 변형된 K-means 알고리즘을 이용해야 한다. [5]는 이등분 K-means 알고리즘(bisecting k-means algorithm)을 제시하여 좋은 결과를 보였다. 따라서 본 논문은 해당 접근방법을 이용하여 주제별 코퍼스를 구축하였다. 다음은 해당 단계를 기술한다.

가) 텍스트 코퍼스 D는 일정 수의 어휘를 갖는 서브텍스트 {d}로 분할된다.
나) d는 벡터 공간(1)로 표현되고, 각 차원은 특정 어휘 t에 대한 tf-idf v값을 갖는다. 여기서 tf-idf v(t,d,D)는 어휘 빈도수 tf(t,d)와 서브 텍스트 빈도수의 역수 idf(t,D)의 곱이다[6].
다) 서브텍스트들은 두 개의 클러스터로 이등분 되고, 각 클러스터 C의 센트로이드 (2)가 계산된다. 여기서 |C|는 해당 클러스터 C의 구성 서브 텍스트 수이다.
라) 각각의 서브 텍스트는 코사인 유사도 (3)에 따라 근접한 클러스터에 할당되고 센트로이드 값은 재계산 된다. 이 단계는 안정화 단계까지 반복된다.
마) 단계 다)와 라)는 작은 수의 서브텍스트들을 포함한 클러스터에 대하여 반복되고, 나머지 서브 텍스트들은 보류된다. 특정 크기의 클러스터가 생성되었을 경우 해당 클러스터는 독립되고, 보류된 서브 텍스트들에 대하여 단계 다)에서 마)까지 반복된다.

$$\vec{t_d} = (v_1, v_2, \ldots, v_n) \tag{1}$$

$$\vec{t_c} = \frac{1}{|C|} \sum_{t_d \in C} \vec{t_d} \tag{2}$$

$$\cos(\vec{t_d}, \vec{t_c}) = \frac{\vec{t_d} \cdot \vec{t_c}}{\| \vec{t_d} \| \, \| \vec{t_c} \|} \tag{3}$$

텍스트 군집화의 결과는 주제별 언어모델 생성에 이용된다. 본 논문의 목적 중 하나는 사용자의 입력에 동적으로 통합되는 언어모델을 제시하는데 있다. 텍스트 군집화를 통해 T개의 주제별 언어모델이 생성되었을 때 (T=|LM|), 사용자 입력 w에 대한 언어모델의 보간 가중치는 식 (5)와 같이 계산되고 식 (4)는 주제별 통합된 언어모델의 결과를 보여 준다.

$$p(w) = \sum_{i=1}^{T} \lambda_i p_i(w) \tag{4}$$

$$\lambda_i = \frac{\log p_i(w)}{\sum_{j=1}^{T} \log p_j(w)} \tag{5}$$

여기서 기존 연구와의 큰 차이점은 언어모델 보간 가중치가 동적으로 연산된다는 점이다. [4]의 경우 보간 가중치는 오프라인 상태에서 미리 연산되어 있고, 특정 영역에 대하여 일괄적으로 적용되는 접근 방법을 취하고 있다. 이는 사용자 입력 영역이 미리 정의되어 있고, 실시간 활용 시점에 해당 입력영역이 무엇인지 알고 있어야 한다. 반면 식 (5)는 사용자 입력에 대한 주제별 언어 모델들의 정규화 된 가중치 값 생성을 통해 다양한 입력 주제들에 대한 대응이 가능하도록 한다.

3. 언어모델 군집화를 통한 동적 언어모델 보간

제시한 동적 언어모델 보간 가중치 연산은 일반영역 음성인식에 적용 가능하다. 그러나 모든 주제별 언어모델에 대한 언어모델값 계산을 요구하는 문제가 있다. 해당 연산에 있어 대상 언어모델의 수를 감소시킬 수 있다면 연산량을 줄일 수 있는 접근 방법이 될 수 있다. 본 장에서는 언어모델 군집화를 통해 이 문제에 대하여 접근한다.

3.1 언어모델 군집화

언어모델 군집화는 주제별 언어모델들을 특정수의 군집들로 분할한다. 유사한 언어모델들을 근접시키기 위해 특정 입력이 요구되는데, 해당 입력에 대하여 유사한 복잡도값[1]을 보이는 언어모델들을 근접시키는 방법을 이용한다.

알고리즘 1. 언어모델 군집화 (D,LM,d)

```
Development Set D = {s_1, s_2 ... s_N}
topic LMs LM = {lm_1, lm_2, ..., lm_T}
for i=1 to E do
    S ← ConstraintRandomSampling(D,LM,d)
    for j=1 to d do
        for k=1 to T do
            pp_j^(k)(i) ← α • PPL(lm_k, s_j) + β • pp_j^(k)(i−1)
        end for
    end for
end for
LMC ← Clustering(LM)
```

알고리즘1에서 개발 집합(development set) D는 문장 s로 구성되어 있다. 여기서 N=|D|이고, D는 언어모델 군집화의 입력 정보가 된다. 군집화 대상 주제별 언어모델 (topic LM) 집합 LM은 각 주제별 언어모델 lm_i로 구성되어 있고, T=|LM|이 된다. 여기서 lm_i는 벡터 형태 (6)으로 표현되고 각 j차원은 무작위로 샘플링 된 문장 집합 S의 구성 문장 s_j에 대한 복잡도 값으로 계산된다. 여기서 d=|S|로 샘플링 된 문장의 개수를 결정한다.

$$lm_i = \vec{p_i} = \left(pp_1^{(i)}, pp_2^{(i)}, \ldots, pp_d^{(i)}\right) \tag{6}$$

알고리즘1에서 3번 줄의 E는 샘플링 횟수를 말한다. 3

1) ppl(perlexity) = $b^{-\frac{1}{N}\sum_{i=1}^{N}\log_b p(x_i)}$ 을 사용하였다.

번 줄에서 10번 줄은 주제별 언어모델에 대한 벡터 공간 모델을 생성한다. 4번 줄의 제약 조건 기반 무작위 문장 부분 집합 추천(Constraint-Random-Sampling)은 개발 집합 D와 언어모델 벡터 공간모델 LM을 이용하여 샘플링 집합 S를 생성한다. 여기서 d는 해당 집합 S의 문장 개수를 의미하게 된다. 본 논문은 제약조건으로 단어의 빈도수와 종결어휘를 규칙으로 사용하였고, 다양한 토픽의 문장 추출을 위해 다중 샘플링 셋을 도출하고, 주제별 LM에 대하여 최소 엔트로피 값을 갖는 샘플링 결과를 선택하였다. 7번 줄은 LM을 구성하는 특정 lm_i의 j번째 차원값이 샘플링된 문장 s_j에 의해 갱신되는 것을 보여준다. 이전 i-1단계의 벡터값과 결합($\beta = 1-\alpha$)되어 해당 차원값은 결정된다.

주제별 언어모델에 대한 벡터 공간이 생성된 이후, 11번 줄의 Clustering은 k개의 언어모델 군집(7)을 생성하게 된다. Clustering은 2장에서 언급한 이등분 K-means 알고리즘을 그대로 사용한다. 따라서 (7)의 M_i의 구성 요소의 개수는 한정되지 않는다. (8)은 언어모델 군집의 한 예를 보여주는데, 여기서 $lm_1^{(i)}$는 M_i의 센트로이드와 가장 근접한 값을 갖는 대표 언어모델로 선택되고, 나머지 언어모델은 해당 유사도 값으로 정렬된다.

$$LMC = \{M_1, M_2, ..., M_k\} \quad (7)$$

$$M_i = \{lm_1^{(i)}, lm_2^{(i)} ... lm_n^{(i)}\} \quad (8)$$

3.2 언어모델 군집화 기반 동적 언어모델 보간

언어모델 군집화 기반 동적 언어모델 보간 기술은 보간 가중치 연산에 참여하는 언어모델 군집들의 개수를 감소시켜 연산 부담을 줄이는 기능을 제공한다. 실시간 입력 문장 w에 대해 $r(i) = lm_1^{(i)}$을 클러스터 (8) M_i의 대표 언어모델로 기술한다면, LMC(7)은 (9)로부터 계산되는 확률값에 따라 정렬가능하다. s(i)를 새로운 LM 클러스터의 인덱스로 정하면 LMC' (10)은 새로운 LM 클러스터 집합이 된다. (10)의 m은 (7)의 k보다 작은 값을 가질 수 있기 때문에 동적 언어모델 보간 연산에 참여하는 주제별 언어모델의 개수를 감소시킬 수 있다 (m < k).

$$\log p_{r(i)}(w) \quad (9)$$

$$LMC' = \{M_{s(1)}, M_{s(2)}, ..., M_{s(m)}\} \quad (10)$$

$rs(i) = lm_1^{s(i)}$을 (1)의 구성 LM클러스터 $M_{s(i)}$의 새로운 대표 언어모델로 기술한다면 각각의 LM클러스터의 보간 가중치 값은 (11)과 같이 계산될 수 있다.

$$\alpha_{s(i)} = \frac{\log p_{rs(i)}(w)}{\sum_{j=1}^{k} \log p_{rs(j)}(w)} \quad (11)$$

LM 클러스터를 구성하는 주제별 언어모델은 (11)에 의해 구해진 보간 가중치 값을 공유하는데, 클러스터의 대표 언어모델과의 벡터 공간 거리에 따라 차등적으로 값 가중치 값이 할당된다. (12)는 이 내용을 기술하고 있다.

$$\alpha_{s(i)} = \sum_{j=1}^{n} \lambda_j^{s(i)}, \ (\lambda_j^{s(i)} > \lambda_{j+1}^{s(i)}) \quad (12)$$

$$p(w) = \sum_{i}^{m} \sum_{j}^{n} \lambda_j^{s(i)} p_j^{s(i)}(w) \quad (13)$$

최종적으로 언어모델 군집화 기반 보간 가중치를 적용한 입력 w에 대한 언어모델 스코어는 (13)으로 구한다. 여기서 주제별 언어모델의 보간 가중치는 입력 w와 언어모델 군집들의 관계성에 따라 동적으로 결정된다. (13)에서 m은 $|LMC'|$ 이고 n은 $|M_{s(i)}|$ 이다. 여기서 n은 언어모델 군집에 따라 가변적이다. 편의상 n이라고 한다. (7)의 k는 $|LMC|$ 이므로, 보간 연산 참여 언어모델의 개수의 감소는 $(m \times n) < (k \times n)$ 으로 설명될 수 있다. 또한 (12)에 참여하는 주제별 언어모델의 개수는 언어모델 군집 내에서 한정 가능하다. 이는 클러스터의 대표 언어모델과 벡터 공간 거리를 이용한 구성 언어모델의 정렬이 가능하기 때문이다. 여기서 한정된 개수를 n^* $(< n)$라고 했을 때 $(m \times n^*) < (m \times n) < (k \times n)$ 가 된다.

4. 관련 연구

동적 언어모델 보간 기술은 언어모델 적응(LM adaptation)과 관련 있다. 사용자 입력 문맥을 이용하여 기본 언어모델을 주제 영역 언어모델들로 적응시키는 형상을 취한다. [7]의 경우 포워드 백워드 알고리즘을 이용하여 사용자 입력 문맥의 우도(likelihood)를 최대화시켜 보간 가중치를 구하는 접근 방법을 제시하였다.

텍스트 군집화 기반 언어모델 적응 기술은 기존 연구 [8]과 [9]에서 진행되었다. 모두 K-means와 유사한 군집화 알고리즘을 이용하여 학습 코퍼스를 분할하여 주제별 언어모델을 생성하는데 사용하였다. [9]의 경우는 tfidf 기반 전통적인 군집화 기술을 이용했고, 반면 [8]은 분할된 텍스트 군집에서 생성된 언어모델의 복잡도 값을 군집화 거리 계산에 사용하였다. 실행 시간시 언어모델 적응은 주제 의존적인 동적 캐쉬 적응 기술을 이용하여 진행되었는데, [9]의 경우는 사용자 이전 문맥을 이용하여 언어모델을 생성하고 이를 주제별 언어모델들과 보간하였고, 반면 [8]은 EM 알고리즘을 이용하여 이전 문맥을 고려한 보간 가중치를 생성하고, 이를 각 주제별 언어모델에 할당하는 접근 방법을 제시하였다.

언어모델 보간 기술의 경우 기존 연구들은 보간 가중치 값 계산 방식에서 차이점을 보인다. 주로 실행 시간에 사용자 입력 히스토리의 우도를 최대화시키기 위해 EM알고리즘을 사용하였다. [3]의 경우는 퍼셉트론 형식의 알고리즘을 도입하여 미리 특정 학습 셋의 음성인식 정확도에 최적화된 보간 가중치 계산 방법을 제시하였으나 미리 연산되어야 되는 문제점이 있었고, 성능 개선 여부도 크지 않았다. 입력 히스토리에 기반을 둔 캐쉬 기반의 언어모델 적응의 경우는 사용자의 입력이 일정하게 진행되어야 하는 영역적 한계가 있다. 따라서 본 논

문은 보간 가중치 연산에 있어 사용자 입력 자체만을 대상으로 하였고, 연산에 참여하는 언어모델 수를 한정하여 연산 효율성을 추구하는 접근방법을 선택하였다.

5. 실험

5.1 실험 환경

실험을 위해 사용한 한국어 음성 인식기는 wFST와 fMLLR[2] 기반 적응 기술을 특징으로 하고 있다. 음성모델(AM)은 1,200시간의 한국어 발성 녹음 자료로 학습되었다. 기본 언어모델(LM)은 17기가바이트의 텍스트로부터 구축되었다. 3-gram으로 구성된 언어모델은 130만 1-gram, 4,210만 2-gram, 4,580만 3-gram으로 구성되었다. 해당 텍스트는 트위터, 뉴스, 게시판, 질의응답 영역으로 구성되어 있고, 웹으로부터 수집되었다.

동적 언어모델 보간 기술의 평가는 음성인식 결과에 대한 N-best 리스코링 접근 방법을 이용한다. N-best는 사용자 입력에 대한 음성인식 결과인 래티스로부터 추출되고, AM/LM 값과 인식 문장의 목록으로 구성된다. 각각의 문장은 AM/LM값을 통합한 인식 결과 값으로 정렬되어 있고, 첫 번째 순위의 인식 문장은 기준 문장으로 선택된다. N-best 리스코링의 절차는 다음과 같다: (1) N-best 목록의 AM/LM 값에서 LM값을 제거한다. (2) 기준 문장에 대하여 주제별 언어모델에 대한 보간 가중치 값을 구한다. (3) 구해진 보간 가중치 값을 이용하여 N-best 목록의 LM값을 다시 계산하고, AM 값과 결합 후 재정렬 한다.

평가셋은 40%의 뉴스, 30%의 SNS, 30%의 게시판 문장으로 구성된 10,000개의 발화를 대상으로 한다. 녹음 환경은 클린 환경과 잡음 환경으로 구성된다. 클린 환경은 조용한 사무실 환경이고, 잡음 환경은 텔레비전, 식당, 지하철 잡음 환경으로 구성된다.

동적 언어모델 보간의 리스코링 평가 척도는 음절 정확도(ACC)로 하고, (14)에 따른다. 여기서 S는 변경 음절의 개수, D는 삭제 음절의 개수, I는 삽입 음절의 개수를 말한다. 다양한 실험 비교는 오류 감소율(ERR)을 이용한다.

$$ACC = \frac{N - D - S - I}{N} \times 100\% \qquad (14)$$

N-best 리스코링에 사용되는 대용량 코퍼스는 다양한 영역으로 구성된 122GB의 텍스트로 구성된다. 주제별 언어모델을 생성하기 위해 이등분 K-means 알고리즘을 통해 모두 58개의 언어모델을 생성했다. 그리고, 고속의 언어모델 보간연산을 위해 해당 언어모델은 LM Trie DB 형태로 구축했다. 이는 ARPA 포맷의 언어모델을 Trie구조로 생성하는 방법이다[3]. 실행 시간시 모든 Trie DB는 메모리에 로딩되어 있고, 언어모델 연산을 수행한다.

2) wFST는 weighted finite-state-transducer의 약자이고, fMLLR은 feature space maximum likelihood linear regression의 약자로 화자 적응의 대표적 기술

추가적으로 텍스트 군집화 접근방법의 타당성 검토를 위해 122GB의 텍스트를 일관되게 분할하여 비주제별 언어모델(non-topic LM)을 생성했다. 그러나, 무작위 셔플링은 진행하지 않아 텍스트 자체의 영역성은 유지된다고 볼 수 있다.

5.2 동적 언어모델 보간 실험

동적 언어모델 보간 실험으로 N-best 리스코링 실험 결과는 표1에 기술되어 있다. 평가셋은 클린과 잡음으로 구성되어 있고, 베이스라인은 음성인식 1-best 결과이다. 텍스트 군집화 기술의 적용 여부에 따라 topic과 non-topic LM으로 실험을 구분하였고, 동적 언어 모델 보간 기술의 적용 여부에 따라 동일 가중치/동적 보간 실험으로 구성했다.

표 1 N-best 리스코링 결과 (ACC %)

		클린	잡음	모두
베이스라인		94.69	84.71	89.7
topic LM	동일 가중치	95.11	85.66	90.38
	동적 보간	95.1	85.66	90.38
non-topic LM	동일 가중치	95.13	85.59	90.36
	동적 보간	95.11	85.64	90.37

텍스트 군집화 기술 기반 주제별 언어모델 구성은 잡음환경에서 좋은 성능을 보였고, 동적 보간 기술은 non-topic LM에서도 어느 정도 효과를 보였다. 그러나 클린 평가셋은 큰 차이를 보이지 못했다. 전체 평가셋 실험의 경우 topic LM의 성능이 좋았으나 큰 성능 개선 차이가 없었고, 동적 보간 실험이 동일 가중치 실험과 차별화를 보이지 못했다.

표 2 언어모델 군집화 결과

	언어모델	f1	f2
c=2 p=4	55	0.9924	0
	29	0.9973	0.9950
	16	0.9982	0.9852
	52	0.9984	0.9891
c=3 p=3	23	0.9870	0
	43	0.9931	0.9781
	4	0.9972	0.9748

5.3 언어모델 군집화 기반 동적 언어모델 보간 실험

언어모델 군집화는 58개의 주제별 언어모델에 대하여 군집화가 진행되었다. 140만 문장을 인터넷 게시판으로부터 추출하여 알고리즘1을 적용하여 주제별 언어모델의 벡터 모델 값을 구했다. 제약 조건 무작위 문장 선택의 제약 조건으로는 6개 이상의 중복되지 않은 어휘수를 포함 조건과 단어 수 14 이상의 문장 조건을 이용했다. 언어모델 벡터 공간의 차원 수는 4,000으로 했다. 즉, 매 군집화 단계에서 4,000문장을 샘플링 했다. 58개의 주제별 언어모델 각각에 대하여 벡터 공간 값이 생성되면 이후 언어모델 군집화를 진행했다.

언어모델 군집화를 통해 58개의 주제별 언어모델은 12개의 언어모델 군집으로 분할되었다. 최소 군집 크기는 2였고, 최대 군집 크기는 10이었다. 표 2는 언어모델 군집의 샘플을 보여준다. 여기서 c는 언어 모델 군집의 번호이고, p는 구성 언어모델의 개수, f1은 군집 센트로이드와의 거리, f2는 군집의 대표 언어모델과의 거리를 말한다. 예를 들면 언어모델 군집화 기반 언어모델 동적 보간은 동적 보간 연산에 있어, c=2에서 lm=55를 선택하여 언어모델 보간 가중치 연산을 하면 c=2의 해당 보간 가중치 값을 구하게 되고, 해당 값을 구성 주제별 언어모델들(55, 29, 16, 52)과 차등 비율에 따라 공유하게 된다. 실험에서는 차등 비율을 표2의 f2값을 이용하여 결정하였다.

표 3 언어모델 군집화 언어모델 동적 보간 실험

	클린	잡음	모두
topic LM	95.12	85.7	90.41
non-topic LM	95.1	85.58	90.34

표3은 언어모델 군집화 기반 언어모델 동적 보간 실험 결과이다. 실험에서는 표1 실험과 비교를 위해 모든 언어모델 군집들을 사용하였다. 표3의 topic LM결과는 표1과 비교했을 때 가장 높은 결과를 보여주고, non-topic LM은 가장 낮은 결과를 보여 주었다. 이는 언어모델 군집화는 텍스트 군집화를 통한 주제별 언어모델 생성을 필요로 한다는 점을 말한다. 큰 차이는 없지만 언어모델 군집화 기반 동적 보간이 좋은 성능을 보인 이유는 동적 보간 연산에 참여하는 언어모델이 언어모델 군집의 대표 언어모델로 한정되기 때문이라 볼 수 있다. 즉, 언어모델 군집화 기술은 입력 문장과 관계가 적은 언어모델이 보간 가중치 연산에 참여하는 것을 회피할 수 있게 한다.

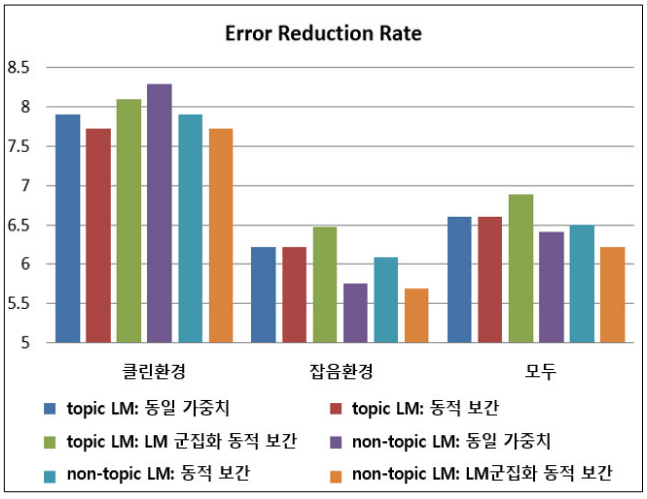

그림 1 실험결과 ERR 비교

그림1은 표1과 표3 실험을 베이스라인에 대한 에러 감소율(ERR)로 표현하고 있다. 클린환경 실험에서는 non-topic LM 동일 가중치 실험이 가장 좋은 반면, 잡음환경에서는 해당 실험의 ERR이 가장 낮다. 통합 환경 실험의 경우 topic LM을 이용한 LM 군집화 동적 보간 실험 결과가 ERR 6.89%로 가장 좋은 성능을 보였다.

5.4 언어모델 군집 축소를 통한 N-best리스코링 실행 시간 단축

표4는 언어모델 군집의 축소(pruning)을 통한 실행 시간 단축 정도를 보여 주는 실험 결과다. 여기서 언어모델 군집은 일정 비율로 축소되고, 언어모델 보간 계산에 참여하는 topic LM의 수는 해당 비율에 따라 감소하게 된다. 이는 사용자 입력에 따라 재정렬 되는 (10)의 LMC'의 m값을 점차 줄여 나간다는 것을 의미 한다. 즉, (13)의 $m \times n$ 개수의 언어모델이 연산에 참여하게 된다.

표 4 언어모델 군집 축소 실험

LM 군집 감소율(%)	topic LM 개수	실행시간 (초)	ACC %
0	58	390.92	90.41
10	52	357.02	90.36
20	47	321.77	90.32
30	42	287.08	90.24
40	36	251.39	90.17
50	27	184.28	90.01
60	22	152.16	89.97

표 4의 실행 시간은 N-best 리스코링에 소요된 시간을 의미한다. 여기서 390.92초는 총 10,000문장에 대한 수행 시간으로 한 문장 당 0.04초의 수행 시간을 보여 준다. 실험결과는 언어모델 군집 축소가 LM군집 감소율 20%에서 0.09%의 ACC저하를 보이고, 17.6%의 실행 시간 단축을 보여주고 있다. 30%의 군집 축소의 경우 0.17%의 ACC저하를 보이고, 26.5%의 실행 시간 단축을 달성 하였다.

6. 결론

본 논문은 대용량 언어모델링 기술을 이용한 음성인식 성능향상에 대하여 기술하고 있다. 텍스트 클러스터링을 통한 주제별 언어모델을 생성하고, 사용자 입력을 고려한 동적 언어모델 보간 기술을 제시하였다. 또한, 동적 언어모델 보간 가중치 연산을 개선하기 위해 언어모델 군집화 기술을 통해 연산에 불필요한 주제별 언어모델을 배제하는 기술을 시도 하였고, 연산 속도 개선을 위한 언어모델 군집 축소 기능에 대한 실험을 보여줬다. 최적의 성능은 주제별 언어모델과 언어모델 군집화 기술을 모두 접목 했을 때 모든 평가셋에 대하여 6.89%의 오류 감소율을 달성하였고, 언어모델 군집 축소 기술을 통해 0.09%의 ACC감소를 통해 17.6%의 실행시간 단축을 달성하였다.

사용자 입력에 적응하고 대용량 주제별 언어모델을 이용한 리스코링을 적용한 실험 결과는 음성인식의 성능개선을 크게 향상 시켰다고 볼 수는 없다. 음성인식기의 N-best 의 스코어에 의존적인 면이 있으며, AM 스코어와 통합하여야만 성능 개선을 볼 수가 있었다. 해당 스코어

의 경우 LM과 고정 비율로 실험할 수밖에 없었는데, 다른 접근 방법을 도출하기는 쉽지 않았다. 또한 LM군집화 동적 보간을 통한 리스코링의 성능 개선에 있어 클린 환경의 개선보다 잡음환경의 개선이 더 큰 이유는 잡음 환경에 따른 N-best의 품질 저하에 대해 LM군집화 동적 보간 기술의 적용이 더 타당하다는 데 있다. 언어모델 군집 축소 실험의 경우는 음성인식 리스코링 성능 개선이 크지 않은 상태에서 0.09%의 ACC감소를 통한 실행 시간 단축이 큰 의미가 없어 보일 수도 있다. 그러나 본 연구의 경우는 다수의 토픽 모델을 통한 리스코링 연산 부담을 감소하는 접근 방법의 하나를 도출한 것에 의미를 두고 있다.

언어모델 보간 기술의 경우 본 연구와 달리 N-best 리스코링이 아니라 1st 단계에서 사용되는 언어모델을 생성하는 기술에 대한 연구 보고가 있다[10]. 향후 고려해 볼만한 연구 방향이라 판단된다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0126-15-1117, 언어학습을 위한 자유발화형 음성대화처리 원천기술 개발]

참고문헌

[1] Y. Zhang, A. S. Hildebrand, and S. Vogel, "Distributed language modeling for Nbest list re-ranking," SIGDAT Int. Conf. Emnlp, Sydney, Australia, July 22-23, 2006, pp. 216-223.

[2] T. Brants et al., "Large language models in machine translation," SIGDAT Int. Conf. Emnlp, Prague, Czech Republic, June 28-30, 2007.

[3] E. Chung et al., "Lattice Rescoring for Speech Recognition using Large Scale Distributed Language Models," ICCL Int. Conf. Coling, Bombay, India, Dec. 8-15, 2012, pp. 217-224.

[4] B. Ballinger et al., "On-demand language model interpolation for mobile speech input," ISCA Int. Conf. Interspeech, Chiba, Japan, Sept 26-30, 2010, pp. 1812-1815.

[5] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," ACM Int. Conf. KDD workshop on text mining, Boston, USA, Aug. 20-23, 2000, pp. 525-526.

[6] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2008.

[7] R. Kneser and V. Steinbiss, "On the dynamic adaptation of stochastic language models," IEEE Int. Conf. ICASSP, Minneapolis, USA, Apr. 27-30, 1993, pp. 586-588.

[8] P. R. Clarkson and A. J. Robinson, "Language model adaptation using mixtures and an exponentially decaying cache," IEEE Int. Conf. ICASSP, Munich, Germany, Apr. 21-24, 1997, pp. 799-802.

[9] R. Iyer and M. Ostendorf, "Modeling long distance de-pendence in language: topic mixtures versus dynamic cache models," IEEE Trans. on Speech and Audio Proc., vol. 7, 1999, pp. 30-39.

[10] C. Allauzen and M. Riley, "Bayesian Language Model Interpolation for Mobile Speech Input," ISCA Int. Conf. Interspeech, Florence, Italy, Aug. 28-31, 2011, pp. 1429-1432.