통계적 언어 모델의 clustering 알고리즘과 음성인식에의 적용

김 우성, 구 명완
한국통신 멀티미디어연구소 음성언어연구팀

A clustering algorithm of statistical langauge model and its application on speech recognition

Woosung Kim, Myoung-Wan Koo
Spoken Language Research Team
Korea Telecom Mulitimedia Technology Laboratory

요 약

연속음성인식 시스템을 개발하기 위해서는 언어가 갖는 문법적 제약을 이용한 언어모델이 요구된다. 문법적 규칙을 이용한 언어모델은 전문가가 일일이 문법 규칙을 만들어 주어야 하는 단점이 있다. 통계적 언어 모델에서는 문법적인 정보를 수작업으로 만들어 주지 않는 대신 그러한 모든 정보를 학습을 통해서 훈련해야 하기 때문에 이를 위해 요구되는 학습 데이터도 엄청나게 증가한다. 따라서 적은 양의 데이터로도 이와 유사한 효과를 보일 수 있는 것이 클래스에 의거한 언어 모델이다. 또 이 모델은 음성 인식과 연계시에 탐색 공간을 줄여 주기 때문에 실시간 시스템 구현에 매우 유용한 모델이다.

여기서는 자동으로 클래스를 찾아주는 알고리즘을 호텔예약시스템의 corpus에 적용, 분석해 보았다. Corpus 자체가 문법규칙이 뚜렷한 특성을 갖고 있기 때문에 heuristic하게 클래스를 준 것과 유사한 결과를 보였지만 corpus 크기가 커질 경우에는 매우 유용할 것이며, initial map을 heuristic하게 주고 그 알고리즘을 적용한 결과 약간의 성능향상을 볼 수 있었다. 끝으로 음성인식시스템과 접합해 본 결과 유사한 결과를 얻었으며 언어모델에도 음향학적 특성을 반영할 수 있는 연구가 요구됨을 알 수 있었다.

1. 서론

연속음성인식 시스템은 기본적으로 음향학적 처리부분과 언어학적 처리부분으로 구성된다. 음향학적 처리 부분은 사람이 발화한 음성으로부터 해당 발음기호들을 찾아내는 부분이며 언어학적 처리부분은 그 발음기호로부터 단어를 만들어 내고,

문장을 만들어 내고, 궁극적으로는 말의 의미를 이해하는 역할을 한다[1]. 언어학적 처리부분은 또 여러 가지 부분으로 구성되지만, 가장 기본이 되는 부분은 발음기호들로부터 해당 단어들을 유추해 내는 부분이며 이를 위해서는 언어모델(Language model:LM)이 사용된다[2]. 즉 언어모델이란 어떤 단어열에 대해 해당 확률값을 부여해 주는데 이는 그 단어열을 사람이 말할 확률을 의미한다. 이는 우리가 사용하는 말이 문법이라는 제약을 갖기 때문에 가능한 것이다. 즉 우리가 사용하는 말은 문법이라는 규칙에 의거하여 발성되며 따라서 이를 반대로 이용하면 어떤 말들이 사용가능하고 불가능한지 예측할 수 있을 것이다. 언어모델은 규칙에 기반한(rule based) 것과 통계적인(statistical) 것의 두가지로 나눌 수 있다. 규칙에 기반한 언어모델은 어떤 전문가가 일일이 문법 규칙을 만들기 때문에 비교적 정확하지만 전문가의 노력이 많이 든다는 단점이 있다. 반면 통계적 언어 모델은 어떤 문법적인 규칙을 사람이 알려주지 않고, 자기가 스스로 학습데이타로부터 그 규칙을 취득하기 때문에 간단하지만, 많은 양의 학습데이타를 요구한다는 단점이 있다. 이 논문에서는 통계적 언어 모델의 단점을 보완해 주는 clustering 알고리즘에 대해 기술하고 이를 음성인식에 적용한 결과를 분석한다.

이 논문의 구성은 다음과 같다. 우선 2장에서는 언어모델에 대해 설명하고 3장에서는 leaving-one-out(LO) criterion[3]에 대해, 4장에서는 clustering 알고리즘에 대해 설명한다. 5장에서는 인위적인 corpus를 생성하여 이 알고리즘을 적용한 결과에 대해 알아보고 6장에서는 음성인식에 적용한 결과를 분석한다. 끝으로 7장에서 결론을 맺는다.

2. 언어 모델(language model)

언어모델은 임의의 어떤 단어열에 대해 그 단어열이 출현할 확률값을 부여한다. 즉 어떤 단어열W가 각 단어 w_i의 열로 이뤄졌다면, $W = w_1, \ldots, w_n$이고, 이의 확률은 다음과 같다

$$p(W) = \prod_{i=1}^{i=n} p(w_i | w_1, \ldots, w_{i-1}) \quad (1)$$

즉 이 식의 의미는 "나는 학교에 간다." 라는 말이 말해질 확률은 문장 처음에 "나는"이라는 단어가 나올 확률, "나는"이라는 단어 다음에 "학교에"라는 단어가 나올 확률, "나는 학교에" 다음에 "간다"라는 단어가 나올 확률, 이 모든 확률값들의 곱이 된다는 의미이다.

이제 이 언어모델이 어떻게 연속음성인식에 쓰일 수 있는지 알아보자. 연속음성인식 시스템은 쉽게 고립단어인식 시스템, 개별적인 단어만 인식할 수 있는 시스템을 연속적으로 적용한 것으로 생각할 수 있다. 즉 연속음성이 개별적인 단어의 연속이므로 개별적인 단어를 연속적으로 인식하게 된다면 연속음성인식 시스템이 되는 것이다. 앞서의 예에서 "나는 학교에 간다."라는 한 문장을 인식하려면 우선 개별적인 단어를 인식해 내야 한다. 우선 문장 처음에 "나는"이라는 단어를 제대로 인식했다면 그 다음에 올 단어를 인식할 때 오기 쉬운 단어 순서는 p(어떤 단어들|나는) 값이 높은 순서가 될 것이다. 그러므로 이 확률값을 이용하면 쉽게 다음 단어를 인식할 수 있다.

그러나 모든 문장에 대해 위 식을 적용할 경우 너무나 많은 파라미터(parameter)들이 존재하므로 이를 단순화시켜서 어떤 단어의 앞에 나타난 M개의 단어만으로

다음의 식과 같이 근사(approximate)시킬 수 있다.

$$p(w_i|w_1, \ldots, w_{i-1}) \approx p(w_i|w_{i-M}, \ldots, w_{i-1}) \quad (2)$$

이런 모델을 실제로 (M+1)-gram 모델이라 하는데 M이 커질수록 파라미터들이 많아지므로 bigram(M=1인 경우)과 trigram(M=2인 경우) 모델이 주로 많이 쓰인다. 즉 bigram의 경우에 어떤 단어가 발성되었다 하면 그 단어 바로 다음에 어떤 단어가 나올 수 있는지 그 확률값을 부여하는 경우이고 trigram의 경우에는 어떤 두 단어 다음에 어떤 단어가 나올 수 있는지 그 확률값을 부여하는 경우이다.

그러나 이를 이용한다 하더라도 실제 음성 인식 시스템의 경우 만약 인식해야 할 단어 수가 20,000개라면 bigram의 경우에 모든 단어마다 각 단어에 대해 다른 단어가 올 확률값을 저장하고 있으니까 $20,000^2 = 4 \times 10^8$개, trigram의 경우에 $20,000^3 = 8 \times 10^{12}$개의 파라미터가 존재하므로 이를 일일이 탐색하기란 어려운 일이다. 실시간으로 음성인식을 수행하는데 있어서 위와 같이 거대한 탐색 공간은 치명적일 수 있다.

이를 보완하기 위해 사용되는 모델이 클래스에 의거한 모델이다. 여기서 클래스란 특성이 유사한 단어들을 묶어 놓은 그룹이 된다. 이렇게 특성이 유사한 단어들을 묶는 과정을 clustering이라고 하며, 이 그룹들을 각각의 클래스라 정의한다. 각 클래스마다 하나의 대표가 있어서 그 대표로 치환하여 학습을 시키고, 반대로 단어를 탐색할 때도 클래스에 의거하여 찾게 된다.

$G: w \rightarrow G(w) = g_w$를 어떤 단어 w를 어떤 클래스 $G(w) = g_w$에 사상시키는 함수라 하고 $|G|$가 클래스의 수를 나타낸다 하면 다음 식이 성립한다.

$$\begin{aligned} p(w_i|w_1, \ldots, w_{i-1}) &\approx p_G(w_i|w_{i-M}, \ldots, w_{i-1}) \\ &= p_G(G(w_i)|G(w_{i-M}), \ldots, G(w_{i-1})) * p_G(w_i|G(w_i)). \end{aligned} \quad (3)$$

만약 $|G|=1000$ 이라면 bigram의 경우에 $1,000^2 + 20,000 = 1.02 \times 10^6$개, 그리고 trigram의 경우에 $1,000^3 + 20,000 = 1.00002 \times 10^9$개의 파라미터를 갖는다. 이는 위의 경우와 비교해 볼 때 매우 축소된 수치이다. 이처럼 이 방법을 이용할 경우 단어를 찾는 탐색 공간을 줄여주기 때문에 음성인식의 성능이 향상된다. 또 이 방법은 클래스에 의거하여 학습을 하기 때문에 동일한 클래스에 속해 있는 단어들은 한 번만 학습을 시키면 되고, 다른 단어들을 학습을 시킬 필요가 없어지므로 학습 데이타를 줄일 수 있는 장점도 있다. 역으로 이야기하면 적은 양의 학습데이터로 학습을 시켰다 하더라도 그 결과가 비교적 robust하다는 의미이다[4].

3. Leaving-One-Out Criterion

실제로 우리가 사용하는 말들은 무수히 많은 단어들로 구성되며, 따라서 이 무수한 단어들에 대해 어떤 단어들을 어떻게 묶어 치환시킬 것인지, 즉 clustering 방법도 매우 다양하며, 그 방법에 따라 음성인식기의 성능도 많은 영향을 받게 된다. 그래서 가장 적합한 clustering 방법을 찾는 것이 요구되며, 이를 자동으로 찾아주는 방법에 대한 연구가 많이 진행되고 있다[5,6,7].

우선 leaving-one-out criterion에 대해 알아보자. 우리가 어떤 패턴인식기의 성능을 평가하고자 할 때는 테스트 데이터가 있어야 한다. 물론 이 테스트 데이터는

학습데이타와는 다른 것이어야 하겠지만 일반적으로 데이터의 양이 충분히 많지 않을 경우에는 전체 수집된 데이터 중에서 얼마를 학습 데이터로 사용하고 얼마를 테스트 데이터를 사용해야 할 지에 따라서 그 평가결과가 다르게 나타날 수 있다. Leaving-one-out criterion을 이용한 방법의 기본 개념은 전체 데이타를 "retained" part T_R과 "held-out" part T_H로 나누는 것이다. 그래서 주어진 T_R로 학습하고, T_H 로 평가를 하는 것이다. 그리고 전체 데이터가 한 번씩 모두 held-out 될 때까지 반복하여 그 결과를 평균한 것을 평가기준으로 사용하는 것이다.

1. 하나의 샘플을 제외시킨다.
2. 나머지 N-1개의 샘플로 학습한다.
3. 제외시킨 한 샘플로 테스트한다.
4. N개의 모든 샘플이 한 번씩 제외되도록 반복한다.

그림 1. Leave-one-out criterion을 이용한 평가 방법

$$F''_{LO} = \sum_{g_1,g_2:N_T(g_1,g_2)>1} N_T(g_1,g_2) * \log(N_T(g_1,g_2)-1-b) + n_{1,T} * \log\left(\frac{b*(n_{+,T}-1)}{(n_{0,T}+1)}\right) - 2\sum_g N_T(g) * \log(N_T(g)-1) \quad (4)$$

식 (4)는 바로 leaving-one-out criterion을 사용한 방법의 optimization criterion으로 사용하게 될 식이다. 여기서 $N_T(g_i,g_j)$은 training corpus T에 (g_i,g_j) 쌍이 나타난 횟수를 말하며 b는 실험적으로 정해지는 상수이다. $n_{0,T}$, $n_{1,T}$, $n_{+,T}$는 각각 training corpus T에서 한 번도 안 나타난 bigram쌍, 1번만 나타난 bigram쌍, 2번 이상 나타난 bigram쌍의 개수를 의미한다.

언어모델의 평가기준이 되는 혼잡도(perplexity:PP)라는 것이 있다. 이 혼잡도는 바로 어떤 단어 뒤에 올 수 있는 평균단어를 의미하는 것으로 이 혼잡도가 낮을수록 평균단어 수가 적다는 의미이고 따라서 음성인식시스템의 입장에서 볼 때는 그만큼 인식이 쉽기 때문에 좋은 언어모델이라고 할 수 있는 것이다[2]. 위의 식은 이 혼잡도를 구하는 식을 leaving-one-out criterion을 적용하여 단순화시킨 것이라고 할 수 있다[4].

4. Clustering 알고리즘

이제는 앞서 구한 optimization criterion을 갖고 실제 어떻게 가장 optimal한 클래스 mapping 함수를 찾아내는가 하는 clustering 알고리즘을 알아보자. 이 논문에서 제시한 방법은 항상 어떤 상태에서 최적의 해를 찾아가는 greedy 알고리즘으로써 초기 조건에 의존하는 local optimal solution을 찾아 준다. 또한 어떤 단어가 어느 클래스에 mapping되는지가 다른 단어의 mapping에도 영향을 주기 때문에 단어의 순서가 매우 중요하다. 여기서는 가장 빈도가 많은 단어부터 빈도가 적은 단어순으로 clustering을 하는 방법을 채택하였다.

Algorithm 1 : Clustering()

start with initial clustering function G
iterate until some convergence criterion is met
{
  for all w in V
  {
    for all g'_w in G
    {
      calculate the difference in $F'''_{LO}(G)$ when w is moved from g_w to g'_w
    }
    move w to g'_w that results in biggest improvement in $F'''_{LO}(G)$
  }
}
End Clustering

그림 2. clustering 알고리즘

5. 호텔 예약 corpus 분석

5.1 호텔 예약 문법과 corpus

한국통신에서는 한일 호텔예약을 위한 음성번역시스템을 개발한 바 있다[8]. 이 시스템을 개발하기 위해서 호텔예약에 필요한 문장들을 수집하고, 이를 분석하여 단어 및 문법을 정의하고, 이것으로부터 인위적인 corpus를 생성하였다. 총 단어수는 총 285개이며, 이를 조합하여 생성이 가능한 문장의 수는 총 245,000,000개가 넘는다. 이는 호텔 예약 시스템에서 인식해야 할 문장 중에서 날짜 부분이 있기 때문이다. 따라서 이를 모두 생성하여 학습시키기란 매우 힘든 일이며 여기에서는 다음과 같은 방법으로 생성하였다. 즉, 일단 1월,..., 12월은 MONTH로, 1일,..., 31일은 DAY라는 이름으로 먼저 휴리스틱(heuristic)하게 clustering을 해 놓은 후 corpus를 생성하였다. 그리고 나서 실제 training과 test에 쓰일 corpus에서는 각기 그 클래스의 소속 단어 중의 하나를 임의로 선정하여 치환하여 놓았다. 그렇게 생성한 corpus에서 4/5는 학습에, 1/5은 테스트에 이용하였다. 그 결과 학습 corpus의 문장 수는 22,235개였고, 테스트 corpus의 문장수는 5,731개였다.

5.2 Initial map에 관한 실험

Clustering 알고리즘 자체가 greedy 알고리즘, 즉 어느 시점에서 가장 좋은 결과를 보이는 곳으로만 수렴을 하는 방법이기 때문에 그 결과가 local optima 일수 있으며, 또 초기 조건에도 많은 영향을 받는다. 여기서도 이를 확인해 보기 위해 초기 조건, 즉 initial map을 몇가지로 주고 테스트 해 보았다. Clustering 알고리즘은 leaving-one-out(LO)방법을 사용하였다. 원래의 clustering 알고리즘에서는 모든 단어들을 하나의 클래스에 할당하고 시작을 하는데 비해, 그 변형으로 uniformly distributed, 즉 maximal 클래스 수에 각 단어들을 균등하게 할당을 한 방법과, random 함수를 이용하여 각 단어들이 임의의 클래스에 할당되게 한 방법, 그리고 heuristic하게 할당을 한 방법을 시험해 보았다. 표 1에 보인 바와 같이 초기 조건으로 heuristic 하게 할당을 한 방법이 하나의 클래스에 할당을 하고 시작한 방법

보다 좋은 결과를 보였으며 다른 방법들은 그리 좋은 결과를 보이지 못했다. 그러나 초기 조건에 의해 cluster된 결과가 영향을 받는다는 사실을 확인할 수 있었다. 참고로 heuristic하게 초기 조건을 주고 학습을 해 본 결과 우리의 문법 자체가 너무 단어들의 클래스가 뚜렷이 구분되고 있기 때문에 heuristic하게 준 클래스들과 모두 유사했다.
표 2는 LO방법의 효율성을 보이기 위해 클러스터링 방법에 따른 결과를 나타낸 것이다. Heuristic하게 클래스를 정의한 경우가 가장 좋았지만 단순히 LO방법만 적용했을 경우에도 유사한 결과를 보였으며, initial map을 heuristic하게 주고 LO를 적용할 경우 약간의 성능 향상을 볼 수 있었다.

표 1. Initial map에 따른 클러스터링 결과

Initial map	Iteration 수	PP
Uniformly distributed	4	7.604733
Randomly distributed	3	7.603224
Clustered as a singleton	3	7.247550
Heuristically clustered	2	7.241603

표 2. 클러스터 방법에 따른 결과

Model	Num. of classes	PP
Class bigram by heuristic	46	7.241490
Class bigram by LO	39	7.247550
Class bigram by LO trained with Heuristic Init. map	45	7.241603

6. 음성인식과의 접합결과

앞서 클러스터링한 결과의 효용성을 알아보기 위해 실제의 연속음성인식 시스템에 이를 적용하였다. 연속음성인식 시스템은 클래스에 의거한 bigram model을 사용하여 N-best 문장을 생성해 내기 때문에 이 클러스터링되는 결과에 많은 영향을 받는다.

실제로 이 heuristic하게 clustering한 경우가 PP는 가장 낮았으나, 음성인식률에서는 heuristic initial map으로 시작하여 LO에 의해 clustering한 경우가 가장 좋은 음성인식률을 보였다. 이는 PP가 낮다고하여 반드시 음성인식률이 좋다고는 볼 수 없음을 시사하고 있으며 이 결과는 이미 다른 연구에서도 언급된 바 있다[9]. 즉 언어모델에서는 어떤 음향학적인 특성은 고려하지 않고 단순히 문법적 특성만 고

려했기 때문에 문법적으로 유사하여 동일한 클래스에 속한 단어들이라 하더라도 음향학적으로도 유사하다면 인식하기가 어렵기 때문에 그런 결과를 나타낸 것이다.

표 3. 언어모델에 따른 음성인식률

Model	Num. of classes	단어인식률(%) Top1(Top5)	문장인식률(%) Top1(Top5)
Class bigram by heuristic	46	93.62(97.57)	75.17(91.22)
Class bigram by LO	39	93.58(97.68)	75.00(91.55)
Class bigram by LO trained with Heuristic Init. map	45	93.66(97.64)	75.34(91.55)

7. 결론

언어모델은 각 단어들에 문법적 정보에 기반한 어떤 확률값을 부여하는 것으로 음성인식에 연계시 많은 장점을 가져다 준다. 그중 통계적 언어모델은 단순히 corpus로부터 스스로 확률값을 결정하기 때문에 일일이 사람이 문법적 정보를 줄 필요가 없는 장점이 있는 반면 많은 양의 학습 corpus를 필요로 한다는 단점이 있다. 이를 보완하기 위해 제시된 모델이 클래스에 의거한 언어모델로 자동으로 적절한 클래스를 찾기 위해 leaving-one-out criterion이 쓰인다. 이 연구에서는 이를 호텔예약시스템에 적용시키기 위해 호텔예약시스템의 문법으로부터 적절히 학습 corpus를 생성해 내고, 이를 위의 방법에 적용, 적절한 클래스를 찾아내었다. 이를 heuristic한 방법과 비교해 본 결과, 비교적 유사한 결과를 보였지만, 문법 자체가 heuristic하게 클래스가 확실히 구분될 수 있을 정도로 단순하기 때문에 heuristic한 결과보다는 좋지 못했다. 그러나 corpus를 인위적으로 생성한 것을 사용하지 않고 우리가 생활에 사용하는 말들을 수집한 corpus로 사용한다면 일일이 heuristic하게 클래스를 주기가 어려운 일이며 따라서 LO방법에 의한 clustering이 유용하게 쓰일 것이다.

또 automatic clustering 알고리즘에서 initial map에 따른 결과를 비교해 보았다. Initial map으로 heuristic하게 분류한 결과를 주고 LO의 방법으로 학습을 한 결과가 가장 좋은 결과를 보였다. 끝으로 클러스터된 결과를 연속음성인식시스템에 연계시켜 본 결과 약간의 음성인식률 향상을 볼 수 있었고, 언어모델에서도 음향학적 특성을 반영할 수 있는 방법이 요구됨을 확인할 수 있었다.

앞으로는 실제 수집한 여러 corpus에 이 알고리즘을 적용해 보고, 이를 바탕으로 연속음성인식시스템을 확장할 예정이다.

〔참 고 문 헌〕

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer. A Maximum likelihood approach to continuous speech recognition. IEEE Tr. on PAMI, Vol. 5, No. 2, pp. 179-190, March 1983.

[2] F. Jelinek. Self-organized language modeling for speech recognition. *Readings in speech recognition,* Morgan Kaufmann Publishers, pp. 450 - 506, 1991.

[3] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[4] Joerg P. Ueberla. An extended clustering algorithm for statistical language models. Technical Report DRA/CIS(CSE1)/RN94/13, Forum Technology - DRA Malvern, Dec. 1994.

[5] H. Ney, U. Essen, R. Kneser. On structuring probablisitic dependencies in stochastic language modelling. *Computer, Speech and Language,* Vol. 8, pp. 1-38, 1994.

[6] R. Kneser and H. Ney. Improved clustering techniques for class-based statistical language modeling. *EUROSPEECH 93,* pp. 973-976, 1993.

[7] Joerg P. Ueberla. More efficient clustering of N-gram for statistical language modeling. *EUROSPEECH 95,* pp. 1257-1260, 1995.

[8] Myoung-Wan Koo, et al., "KT-STS : A Speech Translation System for Hotel Reservation and a Continuous Speech Recognition System for Speech Translation," *EUROSPEECH 95*, pp. 1227-1230, 1995.

[9] Joerg Ueberla. *Analyzing and improving statistical language models for speech recognition,* Ph.D thesis, Simon Fraser University, May 1994.