음성인식 리스코링을 위한 의존관계분석과 상호정보량 접근방법의 비교

정의석○, 전형배, 박전규
음성처리연구실, 한국전자통신연구소
eschung@etri.re.kr, hbjeon@etri.re.kr, jgp@etri.re.kr

Dependency relation analysis and mutual information technique for ASR rescoring

Euisok Chung○, Hyung-Bae Jeon, Jeon-Gue Park
Spoken Langauge Processing Research Section, ETRI

요 약

음성인식 결과는 다수의 후보를 생성할 수 있다. 해당 후보들은 각각 음향모델 값과 언어모델 값을 결합한 형태의 통합 정보를 갖고 있다. 여기서 언어모델 값을 다시 계산하여 성능을 향상하는 접근 방법이 일반적인 음성인식 성능개선 방법 중 하나이며 n-gram 기반 리스코링 접근 방법이 사용되어 왔다. 본 논문은 적절한 성능 개선을 위하여, 대용량 n-gram 모델의 활용 문제점을 고려한 문장 구성 어휘의 의존 관계 분석 접근 방법 및 일정 거리 어휘쌍들의 상호정보량 값을 이용한 접근 방법을 검토한다.

주제어: 음성인식, 언어모델, 구문분석, 상호정보량

1. 서론

음성인식 성능은 최근 상용화가 보편화 될 정도의 성능을 보여 주고 있다. 반면, 검색어 이외의 딕테이션(dictation)과 같은 일반 문장 발성에 대한 인식 성능은 아직 개선이 필요하다고 본다. 본 논문은 언어처리 관점에서 음성인식 성능 개선에 접근한다. 언어처리 기술을 적용할 수 있는 단계는 음성인식 결과에 대한 리스코링(rescoring) 부분이 될 수 있다. 다수의 후보를 생성하는 음성인식 결과는 해당 후보 인식 결과(N-best) 문장과 함께 음향 모델(AM) 값과 언어모델(LM) 값이 결합된 형태의 통합 정보를 제공한다. 여기서 언어모델 값을 다시 계산하거나, 보완하여 N-best 순서를 재정렬 하면 인식 성능을 개선할 수 있다. 대표적인 접근 방법은 n-gram LM 기반 리스코링 접근 방법이 있는데, 히스토리 패턴의 희소성 문제로 인해 대용량 언어모델이 요구되는 문제점이 있다. 본 논문은 해당 문제 해결에 접근하기 위해 문장 의존 관계 분석 기술과 상호 정보량 기반 리스코링 기술을 비교 검증한다.

2. 관련 연구

본 논문의 음성인식 성능 개선을 위한 리스코링 접근 방식(rescoring architecture)은 분산 언어모델 기반 래티스 리스코링 접근 방식[1]을 따른다. 해당 연구에서 리스코링을 위한 음성인식 결과는 래티스(lattice) 형태로 제시되고, 인식 N-best는 래티스로부터 추출된다. 추출된 결과에서 언어모델 값을 다수의 분산된 언어모델 정보로부터 통합하여 재계산 한다. 또한 거리 독립적인 상호정보량(Distance-Independent Mutual Information, DIMI) 언어모델 적용방법도 제공한다. 본 논문의 상호정보량 기반 언어모델은 장거리 문맥 의존 관계(long-distance context dependency) 모델링 기술[3]을 참고하였다.

통계적 의존구조분석 기술은 그래프 기반 접근 방법과 트랜지션 기반 접근 방법이 최근의 연구 경향이다[2]. 본 논문은 구조정보 기반 언어모델(Structured LM) 구현을 위하여 선행 연구에서 트랜지션 기반 의존관계 파서(transition based dependency parser)를 참고하였다. 학습 방법 역시 선행 연구[1]를 참고하여 퍼셉트론 알고리즘(perceptron algorithm)으로 구현하였고, 의존 관계 파싱은 결정적 파싱(deterministic parsing)으로 진행하였다.

3. 의존 관계 분석과 상호 정보량 분석

트랜지션 기반 의존 구조 분석은 스택(stack)과 큐(queue) 구조를 이용한 쉬프트 액션(shift action)과 리듀스 액션(reduce action)으로 구조 분석을 진행한다. 문장 특정 위치의 어절은 스택의 탑(top)과 비교하여 쉬프트 액션과 리듀스 액션을 결정한다. 여기서 쉬프트 액션은 스택으로의 이동이고, 리듀스 액션은 스택의 탑에 위치한 어절과 현재의 어절이 의존관계를 갖고 스택의 탑은 제거되는 과정이다. 현재의 어절은 새로운 스택의 탑과 액션을 결정하게 된다. 액션의 결정은 비교 대상 어절 쌍 각각의 자질 셋(feature set)의 연산 값으로 결정된다. 자질 유형(feature type)은 어절을 구성하는 형태소의 어휘 정보와 형태소 태그 정보의 조합을 이용한

다. 본 논문에서 사용하는 자질 유형은 “스택자질 큐자질 (shift|reduce) weight”형태를 갖는다. 예제 “pc/nc-가/jc 등장/nc-했/xsv R 29”는 스택자질 “pc/nc-가/jc”와 큐자질 “등장/nc-했/xsv”의 경우 리듀스 액션의 스코어가 29라는 의미이다.

자질 셋과 자질 스코어는 퍼셉트론 알고리즘으로 학습하였다. 학습 말뭉치(corpus)는 어절 의존 관계 정보와 형태소 분석 및 태깅 정보가 부착된 형태를 이용하였다. 하나의 문장에 대한 현재까지 학습된 자질 셋과 자질 스코어를 이용하여 트랜지션 파싱(transition parsing)을 진행하였을 때, 모든 의존 관계가 정답일 경우 해당 의존관계에 대하여 자질들을 추출하여 해당 정보를 자질 셋에 추가하고, 해당 자질들 각각에 대하여 1을 증가시켜 주고, 하나의 의존 관계라도 오류의 경우 모든 구성 자질들을 자질 셋에 추가하고 1을 감소 시켜주는 접근 방법을 취하였다. 파싱 과정에서 자질 패턴 값을 활용할 때는 에포크(epoch)의 평균값을 사용하였다.

대략 10~15어절 수준의 2만 5천 구어체 문장에 대한 4fold cross validation방식으로 성능 평가를 진행 하였을 때 다음 표와 같이 의존 관계 정확도가 평균 88.11%을 보였다.

표 1 의존 관계 파싱 성능 평가

epoch	feat. sz	train(acc)	test(acc)
117	3787k	95.24%	87.93%
118	3765k	95.73%	88.56%
119	3840k	96.16%	89.25%
112	3514k	96.19%	86.70%
AVG			88.11%

상호정보량 연산은 PMI(point wise mutual information)값을 사용하였다. 2pair(2P)의 경우 (1)로 계산 되고. 3pair(3P)의 경우는 (3)을 이용한다.

$$pmi(x;y) = \log\frac{p(x,y)}{p(x)p(y)} \quad (1)$$

$$pmi(xy;z) = \log\frac{p(xy,z)}{p(xy)p(z)} \quad (2)$$

PMI값은 대용량 말뭉치로부터 단일 데이터베이스(DB)를 구축하여 접근 가능하다. 본 논문은 17G 말뭉치로부터 PMI DB를 구축하였다. 트라이DB로 구축된 분량은 3pair의 경우 17G크기의 DB였고, 2pair의 경우 3.6G분량의 DB가 되었다. 여기서 크기가 의미가 있는 이유는 ngram LM에 사용된 LM의 크기는 122G말뭉치에서 추출된 58개의 트라이DB로 총 70G분량을 차지하기 때문이다. 실험 결과는 PMI의 접근 방법의 타당성을 말해 준다.

4. 실험

4.1 실험환경 및 실험유형

실험에 사용된 음성인식기는 wFST (weighted finite state transducer)기반 한국어 딕테이션 엔진으로 1,200시간 발성DB로부터 AM을 구축하고, 17G말뭉치로부터 LM을 구축하였다[1]. 래티스 생성을 통한 AM리스코링이 적용된 결과를 베이스라인(baseline)으로 이용하였고, 성능은 음절인식 정확도를 이용하였다. 성능평가에서 사용된 발화 수는 총 10,000발화로 조용한 환경(clean) 잡음환경(noise)으로 동일하게 구성되었다.

실험 유형은 Baseline실험을 제외한 모두 6유형으로 진행 되었다. 표 2의 첫 번째 줄에 기술되어 있는데, 첫 번째 실험은 (A) 파싱 스코어(Parsing score) 실험으로 트랜지션 기반 의존 구조 분석 스코어를 그대로 N-best 리스코링에 적용한 실험이다. (B) HM ngram LM (head modifier Ngram 언어모델) 실험은 말뭉치로부터 파싱 결과의 피수식어-수식어(head-modifier) 쌍에 대한 LM을 구축하고 파싱 결과에서 해당 의존 관계에 대하여 LM 스코어를 도출하는 접근 방법을 시도했다. 그리고, (C) 122G ngram LM 실험은 122G 말뭉치로부터 구축된 70G 분량의 분산 언어모델을 적용한 리스코링 실험이다. 여기서 분산 언어 모델은 입력 문장에 대응하는 다수 언어모델의 최적 가중치 값을 결정하여 언어모델 통합(interpolation)을 통해 리스코링을 진행했다. (D) 17G PMI 2P실험은 3.6G분량의 PMI DB를 이용한 실험이고, (E) 17G PMI 3P 실험은 17G크기의 3P PMI를 적용한 실험이다. 마지막으로 (F) 2P+3P 실험은 2P PMI와 3P PMI를 적절한 비율로 통합하여 진행된 실험이다.

표 2 실험결과 (음절인식률 정확도 %)

실험	clean	noise
Baseline	91.27	74.37
(A) Parsing score	91.34	74.32
(B) HM ngram LM	91.28	74.39
(C) 122G ngram LM	91.89	**75.24**
(D) 17G PMI 2P	91.67	74.91
(E) 17G PMI 3P	91.59	74.48
(F) 17G PMI 2P+3P	**91.94**	74.94

4.2 실험결과 및 분석

실험결과는 표2에 정리되어 있다. (A)파싱 스코어(Parsing score) 실험은 clean 환경의 경우 조금 인식 성능이 향상된 반면, noise 환경의 경우 인식 성능이 감소하는 결과를 보였다. 그 이유는 파싱 자체는 입력 문장을 정문으로 가정하기 때문으로 보인다. 여기서 입력 문장이 비문일 경우 파싱 스코어 값의 신뢰도는 급격하게 낮아지게 된다. N-best의 하위 랭크 인식 결과는 비문의 경우가 상당수이고, 특히 noise 환경 인식결과의 경우 비문의 비중이 clean 환경 보다 더 크다는 문제점이 있다. 성능 개선을 위해서는 비문에 대한 검증 기능이나 오류수정(error correction) 처리 기능이 요구 될 수도 있다고 본다.

(B)HM ngram LM(head-modifier Ngram 언어모델) 실험

은 성능 개선은 미비한 결과를 도출했다. 해당 이유 역시 파싱 스코어 실험의 문제점과 동일하다는 판단이다. 즉, N-best에 존재하는 비문으로 인한 의존 구조 분석 오류가 HM ngram LM과 해당 문장의 불일치성을 발생한다고 말할 수 있다.

(C)122G ngram LM 실험은 clean의 경우는 7.1%의 오류 감소률(error reduction rate, ERR)을 보였고, noise의 경우는 3.3%의 ERR을 보였다. 대용량 n-gram LM기반 리스코링은 clean환경, noise환경 모두 좋은 성능을 보였다. 이는 대용량 언어모델이 n-gram 희소성 문제를 어느 정도 해결해 준다는 장점 때문일 수 있다.

(D)17G PMI 2P실험은 4.5%의 clean ERR과 2.1%의 noise ERR로 (C)ngram LM 리스코링 실험의 결과에는 못 미치는 성능을 보였다. (E)17G PMI 3P 실험의 경우는 2P 성능보다도 결과가 좋지 않았던 반면, (F)2P+3P 실험의 성능은 noise의 경우 ngram LM의 성능보다 ERR이 작으나, clean의 경우는 7.6%의 ERR로 가장 좋은 성능을 도출했다. 이는 상호정보량 기반 접근 방법의 효용성을 보여 준다. 리스코링용 자원 크기에 제약이 없다면 n-gram LM과 통합하여 사용하면 좋은 결과를 얻을 수 있으리라 본다.

5. 결론

본 논문은 structured LM의 구현을 위한 접근 방법들을 검토했고, 문장 의존 관계 분석 접근 방법의 활용은 부적절하다는 결론에 도달했다. 반면 상호정보량 기반 접근 방법은 기존 ngram LM 리스코링 성능보다 clean환경의 경우 더 좋은 결과를 보여줬다. 30% 수준의 자원(resource) 양을 이용하여 좋은 결과를 도출한 점은 의미가 있는 결과라 판단된다.

참고문헌

[1] E. Chung, H.B. Jeon, J.G. Park and Y.K. Lee, Lattice Rescoring for Speech Recognition Using Large Scale Distributed Language Models, COLING 2012, 2012.

[2] Y. Zhana and S. Clark, A Tale of Two Parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search, Computational Linguistics, 2008.

[3] Z. Guodong, Modeling of Long Distance Context Dependency, COLING'04, 2004.