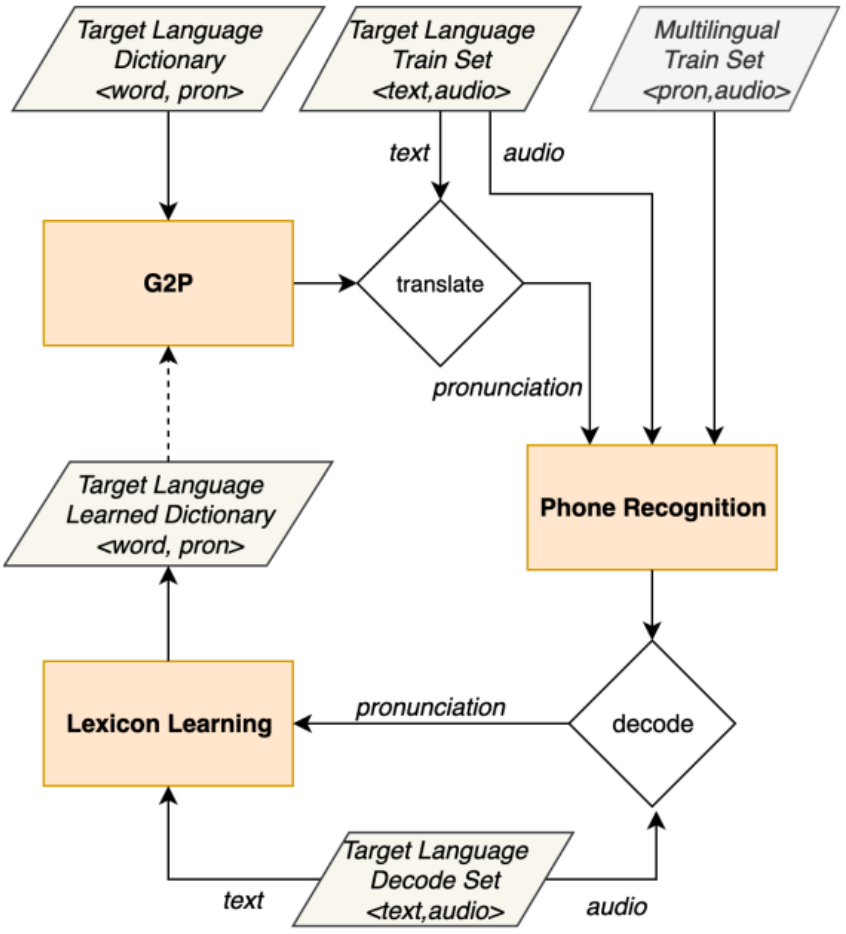
Improving grapheme-to-phoneme conversion by learning pronunciations from speech recordings

[Amazon Alexa, TTS Research]



I. 논문 목적

- CS-ASR과 개체명 인식의 낮은 성능을 보완하기 위한 기존의 데이터 증강은 TTS와 Audio Splicing에 의존하는데 이를 뛰어넘는 text-based speech editing model을 제안
- G2P 시스템의 PER을 다양한 언어와 다양한 데이터에서 감소

II. 기존 연구(G2P)

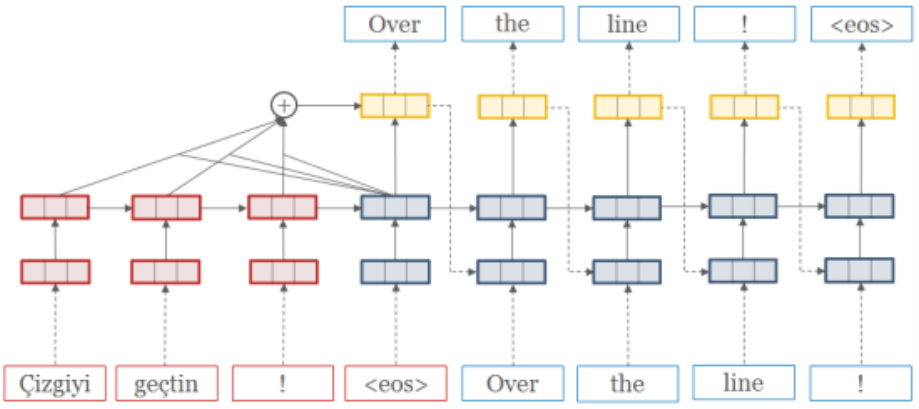
- 정의 : Grapheme to Phoneme): 자소(문자) 시퀀스 ⇒ 음소(음향) 시퀀스 ex) 김밥 → 김빱
- 활용: TTS, ASR에서 어휘 이외 단어 발음 디코딩, 발음사전 일반화
- 문제점: 발음 사전 크기에 의존 → 시간/비용 소요 ↑, 음성학 전문가 이상의 목표 언어 지식 필요
- 전통적 데이터 기반 G2P 접근법
    - Decision Tree
    - HMM(Hidden-Markov Model)
    - grapheme/phoneme Joint N-gram
    - WFST(Weighted Finite-State Transducer)
- ANN: {LSTM, Transformer-based, ... }
    - 전통적 데이터 기반 접근법보다 성능 우수
- low 리소스 G2P 대안 방법
    - 다국어 G2P 시스템
        - 목표: 언어 간 차이 최소화, 대규모 발음 사전에 대한 의존 최소화, high→low 리소스 언어에 대한 성능 전이
    - 텍스트 기반 비지도 선학습
    - 텍스트 기반 데이터 증강
- 자동 발음 학습
    - 제로샷 상황에서 음성 샘플로 새 언어 음성목록(지식) 생성
    - 보편 음소 인식을 사용한 자동 발음 전사
- 음성 데이터로 기존 G2P 시스템을 반복/수정/보충

III. 제안 알고리즘

- 목표: 음성 파일로부터 어휘에 없는 발음 예시를 학습시켜 G2P 모델 발전
    - 다국어 transformer 기반 G2P 모델 사용 → 언어간 지식 전이, 자동 발음 전사

- 학습 과정 요약
  - 처음 본 목표 언어 발음사전(<단어, 발음>), 음성 말뭉치(<텍스트, 음성>)으로 baseline G2P 모델 학습
  - baseline G2P 모델로 목표 언어 음성 디코딩
  - 목표 언어 <발음, 음성> 쌍으로 증강된, 다국어 데이터 <발음, 음성>쌍 사용하여 음소 인식기 학습
  - 문장 수준에서 목표 언어 음성 데이터 디코딩
    - 음소 인식기는 문장 수준에서 작동하지만 단어 경계에 대한 지식 無

    → 디코딩된 발음 시퀀스 - 문자 시퀀스 align(조정)
  - 단어 경계 발견, 단어 수준 발음사전 학습
- **(1) G2P 변환**
  - Open-NMT(Neural Machine Translation) transformer encoder-decoder 아키텍쳐 기반
    - 기존의 NMT 구조

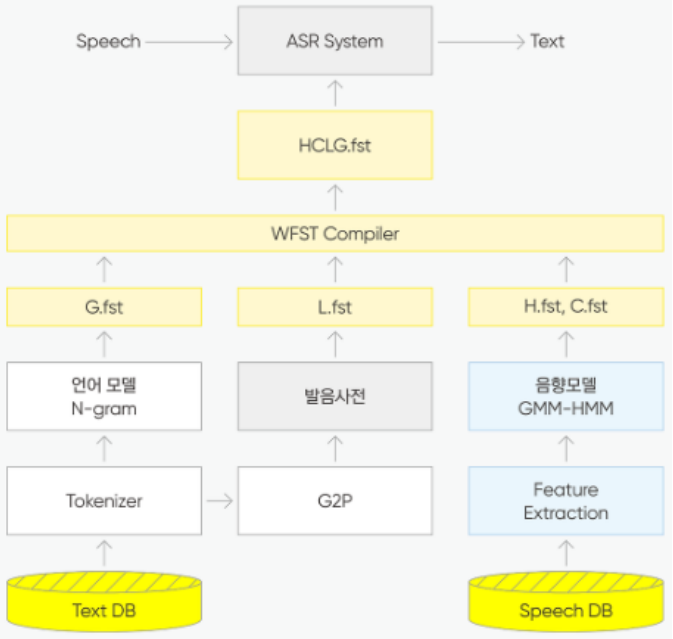


      - 사진에서의 입력(빨강), 목표(파랑)
      - 입력 단어가 단어 벡터에 매핑 후 입력 RNN에 입력 → <eos> 발견 시 목표 RNN 초기화 → 각 목표 step에서 입력 RNN에 어텐션 적용 & 현재 은닉상태 결합(+) → 다음 단어 예측 → 목표 RNN으로 피드백
      - 위 모델에서 훈련/테스트 효율성에 우선순위, 모델 모듈성과 가독성 유지, 연구 확장 지원 → OpenNMT toolkit
    - 모델 구조
      - 인/디코더 모두 레이어 6개, 어텐션 헤드 8개

        - 은닉 FFN 노드 2048개, 임베딩 크기 512, 드롭아웃 비율 0.1
        - adam 옵티마이저, noam 스케줄러, warmup steps 8000
    - level & I/O: 단어 수준, 언어 태그가 붙은 자소 시퀀스 ⇒ X-SAMPA 에서 정의된 음소 시퀀스
        - IPA → SAMPA(Speech Assessment Methods Phonetic Alphabet → 기계 가독성과 국제적 협력)

ɑ	script a, open back unrounded vowel, card. 5, Eng. *start*	A
æ	ae ligature, near-open front unrounded vowel, Eng. *trap*	{
ɐ	turned a, open schwa, Ger. *besser*	6
ɒ	turned script a, open back rounded vowel, Eng. *lot*	Q
ɛ	epsilon, open-mid front unr.vowel, card. 3, Fr. *même*	E

  - 세부 과정: G2P 모델을 다국어 발음 말뭉치에 대한 선학습 → baseline G2P 모델을 처음 본 목표 언어 <단어, 발음> 쌍에 대해 최대 20k step으로 FT
- **(2) 음소(phone) 인식**
  - 목표: 음성 파일에 음소 시퀀스에 상응하는 주석 달기
  - 방법
    - baseline G2P로 목표언어 학습셋(<텍스트, 음성>)에 대한 발음 사전 생성
        - 목표언어 학습셋 ⊂ 발음 사전이 있는 대규모 다국어 음성 데이터셋
    - kaldi 이용하여 MFCC(Mel Frequency Cepstral Coefficient) 추출 → 단일/삼중음소 HMM-GMM 모델 초기화
        - Kaldi의 ASR 훈련 ← 언어 & 발음 & 음향 모델 필요



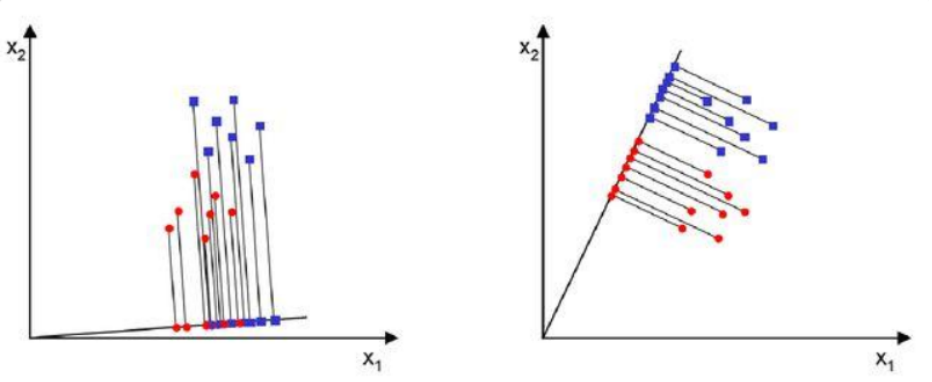
    - 음향 모델 입력값: 음성에서 문자와 관련된 부분
- MFCC(Mel-Frequency Cepstral Coefficient)
    - 프레임 단위로 수행(25 ms, 10ms 이동)
    - 데이터 추출, noise(dithering) & offset & windowing 전처리

    → FFT(Fast Fourier Transform) → Power 스펙트럼 계산

    → 각 mel bin의 에너지 계산 → 로그 계산 → 스케일링(Cepstral lifting)
- PLP(Perceptual Linear Predictive technique)
    - 인간 청각 인지 과정을 모방하여 특정 주파수 대역에서 민감 반영하여 특징 추출
        - ≠ LPC(Linear Prediction Coefficient): 신호 선형 예측 중점
- VTLN(Vocal Tract Length Normalization)
    - MFCC나 PLP 계산 과정에서 사용 가능
        - 중심 주파수를 이동시켜 성대 길이 정규화(ex) 남녀 음높이 차이 ↓)
        - [low-freq, high-freq] → [vtln-low, vtln-high]
            - 0 <= low-freq <= vtln-low < vtln-high < high-freq <= nyquist

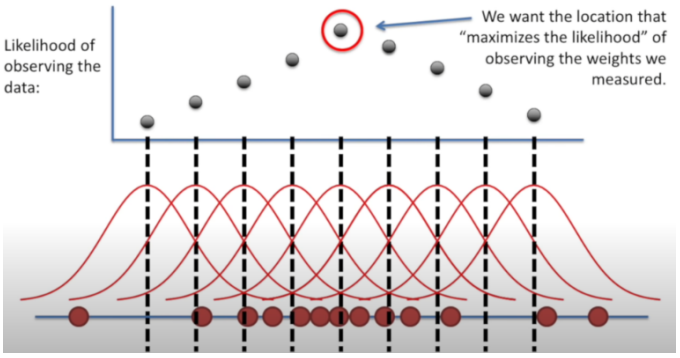
- HMM-GMM(Hidden Marcov Model with Gaussian Mixture Model)
    - HMM: 음성 신호를 일련의 상태로 모델링하여, 상태 간 전환 확률 기반으로 음성 데이터 순서 추정
    - GMM: 각 음성 특징을 여러 가우시안 분포의 조합으로 표현
    - kaldi: HMM-GMM 기반으로 음소 추정, WFST(Weighted Finite-State Transducer) 사용해 주어진 음성을 단어열로 디코딩
        - determinization: 여러 출력값 가능성을 유한한 상태로 결정 짓는 과정

- LDA & MLLT 적용
    - LDA(Linear Discriminant Analysis, 오른쪽)

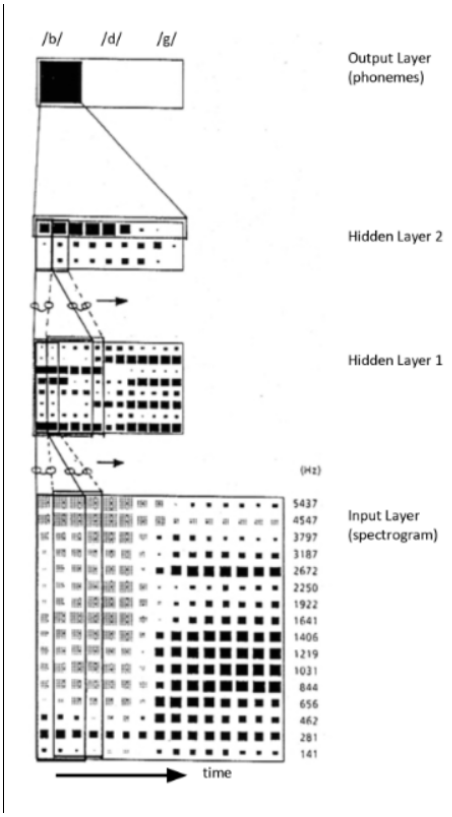


출처: https://slidesplayer.org/slide/16218884/

        - 저차원 공간으로 투영, 분류 특화(클래스 간 분산 ↑, 클래스 내부 분산 ↓)
            - ≠ PCA(왼쪽, 최대 분산 차원으로 축소 → 손실 정보 최소화)
    - MLTT(Maximum Likelihood Linear Transform, 최대우도선형변환)

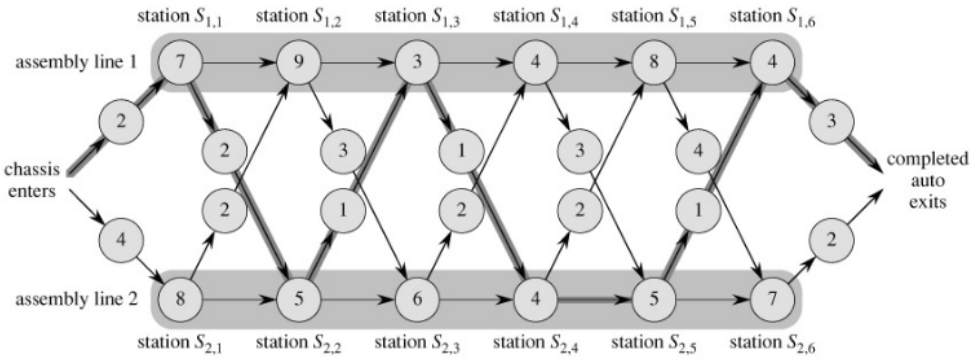


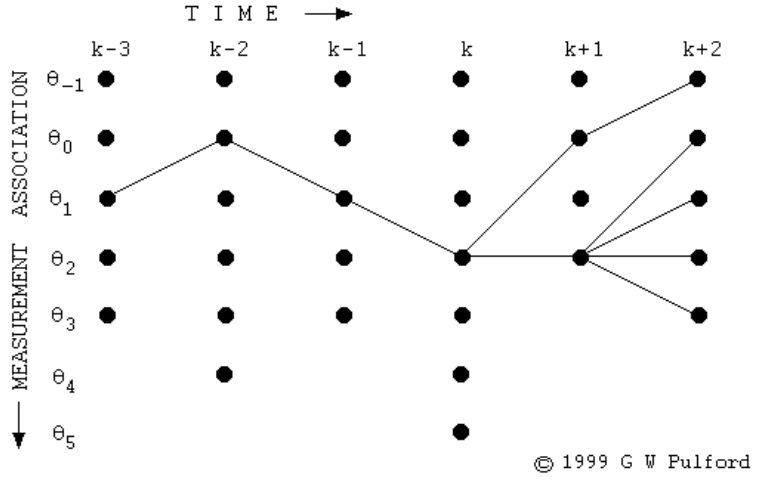
    - 우도: 관측치가 특정 확률 분포에서 왔을 확률
    - 최대 우도: 관측값들의 총 우도가 최대가 되는 분포
    - 최대 우도 가정한 선형 변환 → 음성인식에서 음성 데이터를 음향 모델에 맞게 변환하는 작업
- fMLLR로 화자 적응형 학습 진행
    - fMLLR(Feature space Maximum Likelihood Linear Regression)
        - 신호처리에서 화자 적응형으로 feature 변환하는 방법
        - 화자마다 feature가 너무 상이하여 보편적인 feature를 추출하기 어렵기 때문에 화자간 차이를 최소화하여 발음과 직접적으로 관련된 feature만 추출하기 위함
- 위의 feature & alignment로 TDNN 모델 학습
    - 음성 모델에 언어 식별자 포함 X → 다국어간 음성 일반화, baseline G2P의 잘못된 디코딩(노이즈) 극복 가능
    - TDNN(Time Delay Neural Network)



        - sliding window 방식으로 일정한 길이의 데이터를 일정 부분씩 겹치도록 전처리 한 후 입력값을 넣어주는 네트워크
    - 학습셋의 음소 수준 전사를 학습한 5gram 언어 모델로 음성 샘플 디코딩
        - 언어모델은 G2P 결과 & 목표 언어 음소 배열론 이용하여 디코딩
    - 음향모델은 다국어 음성 샘플을 통해 학습한 보편 음소 지식에 따른 디코딩
- **(3) 발음사전(lexicon) 학습**
    - 디코딩된 음소 시퀀스를 자소 시퀀스에 FA(강제 할당)
        - 문장 수준에서 음성 디코딩 → 음소 인식 시스템이 단어 경계 없는 음소 시퀀스를 출력하는 문제 해결 위해
    - 디코딩셋 어휘에서 각 단어마다 HMM 모델 정의
        - HMM모델: skip 연결이 있는 좌-우 위상 따름, 각 상태는 단어의 자소에 해당
    - 주어진 음소에 대한 각 상태 할당 확률은 음소 공간에서의 이산확률분포 따름
        - 이 분포는 균등분포로 초기화되고 상태/음소에 걸쳐 연결됨
    - 디코딩셋의 각 문장: 단어수준 HMM을 concat하여 문장수준으로 형성

- Viterbi 알고리즘으로 디코딩된 발음 시퀀스로부터 최적의 문장(자소 시퀀스) 선택





  - 은닉 상태의 최적 시퀀스 찾기 위한 동적 프로그래밍 기법(각 단계마다 그 전 단계의 결과값을 이용)
  - ~= Assembly-Line Scheduling; 최소 비용의 공장 생산 라인 도출
  - k에서 각 노드들까지의 최적값을 역추적(k-1)하여 한 경로만 남김 → 각 단계 반복하며 불필요한 경로 삭제 → 최종 경로
- pomegranate 확률모델로 HMM 모델 최적화
  - pomegranate: HMM모델 & 확률분포를 쉽게 다룰 수 있는 라이브러리
    - DenseHMM, SparseHMM, GMM, kmeans, markov_chain, factor_graph, baysian_network, bayes_classifier + 분포(베르누이, 감마, 정규, 균등, ...)
- 디코딩셋 어휘의 각 단어에 대한 디코딩된 발음 수집하여 발음 사전을 생성

- k(<단어,발음>을 발음 사전에 포함시키기위한 단어-음소 쌍 관찰 최소 임계값) 정의
  - 높을 수록 발음 사전 목록 수 감소, 발음에 대한 신뢰도 증가

IV. 실험 및 결과

- 리소스 상황
  - low: 500 최빈단어 (⊂ 2M 문장 ⊂ 다국어 말뭉치 C4) 의 사전
- 다국어 G2P 모델 선학습 조건 및 언어: 1M steps, <단어, 발음> 4.5M 쌍, 17개국어
  - → 목표 언어(영어, 불어, 덴마크어, 폴란드어, 터키어) <단어, 발음> seed 데이터에 대해 FT → FT 된 G2P 시스템으로 목표언어 음성에 대한 발음 생성 = 목표 언어 발음사전 학습
    - 음성 데이터: 목표 언어 9~27hr + 16개국어(17개국 - 목표언어 제외) 6k 화자 다국어 음성 165 hr 의 일부(언어마다 차이)
  - → 학습된 발음 & 목표언어 seed<단어, 발음> 쌍으로 다시 FT → 학습에서 보지 못했던 단어 토큰에 대한 테스트(평가 항목: PER, WER)
- **(1) 발음사전(lexicon) 학습**

	Average (5 languages)		English					
k	PER (Learned)	PER (G2P)	PER (Learned)	PER (G2P)	Num Words	Better	Worse	Same
1	12.99%	**12.53%**	**15.32%**	17.31%	26557	4360 (16.42%)	2431 (9.15%)	19766 (74.43%)
2	**9.27%**	10.25%	**11.13%**	13.45%	12271	1328 (10.82%)	359 (2.93%)	10584 (86.25%)
4	**7.86%**	8.55%	**9.15%**	10.60%	5943	364 (6.12%)	90 (1.51%)	5489 (92.36%)
6	**7.33%**	7.80%	**8.57%**	9.38%	3962	153 (3.86%)	56 (1.41%)	3753 (94.72%)
8	**6.93%**	7.23%	**8.19%**	8.64%	3009	72 (2.39%)	40 (1.33%)	2897 (96.28%)

  - baseline G2P 발음사전 v. learned(논문 방법) 발음사전 ← k & 언어(영어, 5언어 모두)에 따른 PER 비교
  - k=1 일 때 5언어 평균 제외하고 모두 baseline보다 성능 우수
    - 폴란드어, 터키어의 데이터 양 부족 문제로 추정
  - baseline과 비교하여 same, better, worse로 나눠 비율 측정 → 대부분은 same이지만 better > worse

- **(2) G2P 변환**
  - 학습된 발음 사전 이용하여 다국어 G2P 시스템 재학습

- baseline이 생성한 발음사전으로 FT된 G2P v. 제안된 발음사전으로 FT된 G2P의 k에 따른 PER, WER, PERR(5언어 평균)

System	k	PER	WER	PER Rel. Reduction
Baseline	-	13.02%	50.16%	-
Learned	1	**10.64%**	**45.39%**	**-18.32%**
	2	11.02%	46.06%	-15.36%
	4	11.47%	46.50%	-11.94%
	6	11.69%	47.03%	-10.23%
	8	11.86%	47.53%	-8.89%

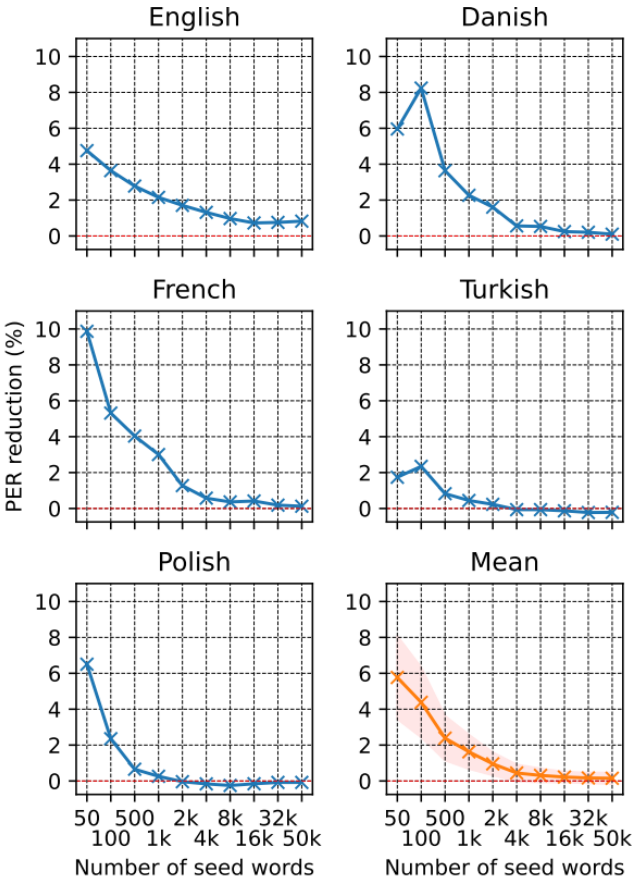
  - k=1 일때(사전 목록 수 최다) 최고 성능
    - 이전 결과는 k=1만 성능 저하
      - low 리소스 상황에서는 G2P가 데이터 양 증가할수록 성능 증가 추정
  - 음소 인식기가 G2P 오류 정정, 올바른 발음 예시 검증
- k=1~2 일때 각 언어 baseline G2P v. learned G2P의 PER, PERR

Language	Baseline	$k = 1$	$k = 2$
English	19.20%	**16.42%** (-14.48%)	17.01% (-11.38%)
Danish	20.57%	**16.94%** (-17.67%)	17.76% (-13.66%)
French	12.13%	**8.09%** (-33.28%)	8.86% (-26.93%)
Turkish	9.27%	8.44% (-8.90%)	**8.41%** (-9.28%)
Polish	3.95%	3.30% (-16.58%)	**3.07%** (-22.28%)
Average	*13.02%*	***10.64%** (-18.32%)*	*11.02% (-15.37%)*

  - 폴란드어, 터키어에 대해서만 k=2, 나머지는 k=1이 최고성능
    - G2P가 이미 이 언어들에 대해 ER이 낮기 때문으로 추정
    - 폴란드어, 터키어는 다른 언어들보다 자소-음소 대응이 간단
    - 데이터의 양(단어사전 목록 수)보다는 신뢰도를 높이는 방안(k를 낮추는 방안)이 더 효과적

- **(3) 기반(seed) 데이터 양**

- k=1일 때 seed 데이터 양에 따른 baseline v. learned PERR



  - 여기서는 양수가 성능 증가 의미
  - 1K 개 단어 이하일 때 성능 증가, 평균 PERR 2~6%
  - 50일 때 덴마크, 프랑스, 폴란드어 PERR 6~10%
  - 2K~(middle ~ high 리소스)에서 폴란드, 터키는 1% PER 증가(성능 저하), 나머지는 성능 향상

- **(4) 자가 학습 반복**
  - k=1, seed 데이터 100 or 500 같은 데이터로 반복 학습 1~5회 시 영어, 덴마크어의 PER, PERR

Language	System	100 seed words		500 seed words	
		PER	Rel. Red.	PER	Rel. Red.
English	Baseline	22.26%	-	22.02%	-
	Iter 1	18.44%	-17.14%	17.62%	-11.97%
	Iter 2	17.71%	-3.96%	17.16%	-2.61%
	Iter 3	17.71%	0.00%	16.97%	-1.14%
	Iter 4	17.69%	-0.14%	**16.87%**	-0.59%
	Iter 5	**17.59%**	-0.54%	17.03%	0.95%
Danish	Baseline	39.60%	-	20.63%	-
	Iter 1	30.56%	-22.81%	**17.29%**	-16.19%
	Iter 2	30.26%	-1.00%	17.31%	0.12%
	Iter 3	29.81%	-1.50%	17.35%	0.23%
	Iter 4	**29.56%**	-0.82%	17.46%	0.63%
	Iter 5	29.56%	0.00%	17.43%	-0.20%

- 여기서는 음수가 성능 증가 의미
- 반복 안 한 것 보다 반복 하는 것이 성능 ↑
- 회당 반복 영향력은 횟수 거듭 시마다 ↓

V. 결론 및 향후 과제

- 결론
  - low 리소스 상황에서는 k ↓, 사전 목록 수 ↑ (신뢰도 ↓)수록 성능 ↑
  - G2P가 오류를 많이 발생시키는, low 리소스 상황에서 발음학습 시스템 영향력 ↑
  - 반복학습 → 성능 ↑
- 향후 과제
  - high 리소스 상황에서의 자동 발음 학습
  - 고유 명사, 외래어(loan), 도메인 특수토큰 발음 학습
  - 음성 데이터 양이나 유형에 따른 차이 실험
  - 다중화자 음성 녹음 실험
  - SSL(자가지도학습) 상황에서의 선학습

VI. 참고 문헌

- G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, “OpenNMT: Open-source toolkit for neural machine translation,” in Proc. ACL, 2017, pp. 67–72.
- J. C. Wells, “Computer-coding the IPA: a proposed extension of SAMPA,” Revised draft, vol. 4, no. 28, p. 1995, 1995.
- L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” in Proc. NAACL, 2021, pp. 483–498.
- https://ratsgo.github.io/data structure&algorithm/2017/11/14/viterbi/
- https://blog.naver.com/urisystem72/222838442994
- https://blog.naver.com/d_f_company/223246019814
- https://newsight.tistory.com/186
- https://kaldi-asr.org/doc/model.html
- https://blog.naver.com/QD/222623415046