

딥러닝 스터디

Recurrent neural network based language model

김제우

목차

1. 제목 보고 흐름 예상하기
2. 모델 구조 Main Feature 보고 예상하기
3. 논문 읽기
4. 정리

제목 보고 감 잡기

- Recurrent neural network based language model
- RNN기반 Language model

예상할 수 있는 것

- RNN에 대해서 전제하고 있겠구나
- RNN에 기반한 Language model 이니까 language model로의 역할을 하겠다.
 - 문장의 '자연스러운 정도'를 확률로 이해하는 것

모델 구조 main feature 보고 예상 해보기

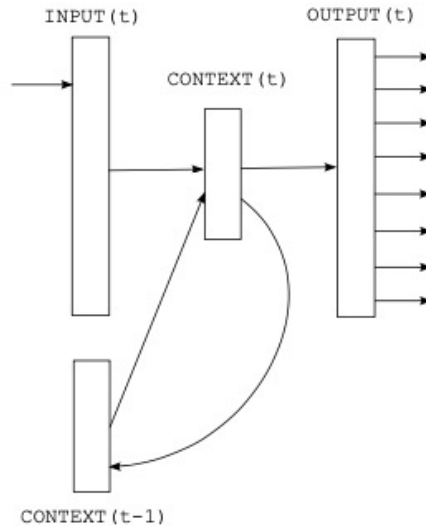


Figure 1: Simple recurrent neural network.

예상할 수 있는 것

- $context(t)$ 현재상태가 이전 $context(t-1)$ 에 영향을 받음.
- 그냥 RNN구조

저자 & 출시 시기

- Tomas Mikolov - brno
- Martin Karafiat - brno
- Lukas Burget - brno
- Jan 'Honza' Cernocky - brno
- Sanjeev Khudanpur - johns hopkins
- 2010년

예상할 수 있는 것

- Word2Vec 보다 먼저 나왔음. -> 단어의 분산 표현에 대해서도 연구할까?
- w2v이전의 이야기를 할 것이기 때문에 통계학적 모델 같은 것들에 대한 사전지식이 필요할지도?
- Tomas Mikolov - w2v의 저자
- 체코에 브루노 대학과 존스 홉킨스 대학 전자 컴퓨터 공학과가 같이 작업함.

Abstract

- 음성인식을 위한 RNNLM이 어플리케이션과 함께 공개되었다.
- 그 결과는 RNN LM들을 섞어서 50%의 perplexity를 감소시켰다는 것을 보여주고, 이전 language model들에 비해 최신의 성능이다.
- 음성인식 실험들은 모델들을 같은 양의 데이터로 훈련시켰을때보다 월스트리트 저널 task 에서 18%의 단어 에러 비율 감소시켰고, 더 어려운 데이터인 NIST RT05에서는 5%의 감소가 있었으며, 이것은 RNNLM보다 더 많은 데이터로 훈련시킨 이전 모델들 보다는 좋았다.
- 우리는 연결주의 언어 모델이 높은 계산(훈련) 복잡성을 제외하고 표준 n-그램 기술보다 우수하다는 것을 시사하는 충분한 경험적 증거를 제공한다.

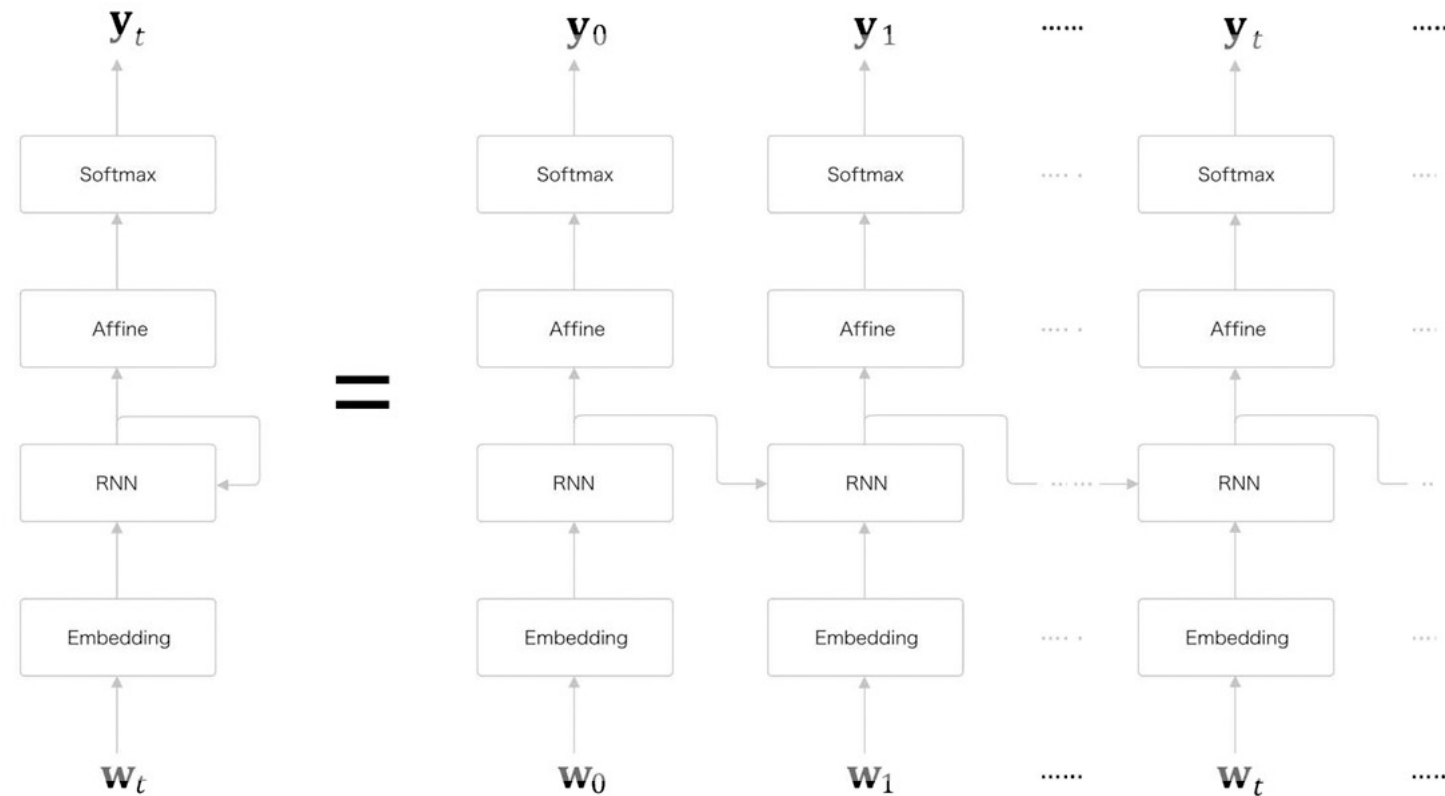
- 순차적인 데이터의 예측은 AI와 ML의 많은 핵심 문제들로서 consider(여겨진다).
- 확률적 언어모델링의 목표는 주어진 컨텍스트의 원본 데이터안에서 다음 단어를 예측하는 것이다. 따라서 우리는 언어모델을 구축할 때 순차적인 데이터 예측 문제를 함께 다룬다.
- 그럼에도 불구하고, 많은 확률 모델들을 얻으려는 시도들은 언어 도메인을 위한 매우 정확한 접근들을 포함한다.
- 예를 들면 자연어문장이 parse tree(어원 트리)로 묘사될 수 있다거나, 또는 단어의 morphology(형태론), 문법, 의미를 고려해야한다는 가정이다.
- n-gram 통계에 기초한 가장 널리 사용되고 일반적인 모델조차도 언어는 문장을 형성하는 원자 기호 -단어-의 시퀀스로 구성되며, 문장 기호의 끝부분이 중요하고 매우 특별한 역할을 한다고 가정한다.

- 만약 언어 모델링에서 간단한 n-gram 모델을 넘어서는 거대한 진보가 있었는지 의문이 들만한 요소이다.
- 만약 우리가 이 진전을 측정해야한다면, 모델의 성능에 의해 시퀀셜 데이터에 대한 더 나은 예측을 위한, 답은 아마도 기여할만한 진전은 캐시모델과 class기반 모델소개에 의해 달성되었을 것이다.
- 다른 여러 기술들이 제안되었지만, 그것들의 영향은 캐시모델이나 클래스기반 모델들에 영향을 끼치지 못했다.
- 만약 우리가 강화된 언어모델링 기술들의 성공률에 대해 그들의 연습용 어플리케이션을 통해 측정해야한다면, 우리는 더 회의적이다.
- 실제 음성인식이나 기계번역시스템을 위한 언어모델들은 거대한 데이터에 의해 만들어졌고, 더 많은 데이터만이 필요하다는 믿음을 준다.
- 모델은 연구들에 의해 나왔고, 더 복잡하려는 경향성과, 아주 제한적인 학습데이터에 의해서만 사용한 시스템에서 잘 돌아간다.
- 사실, 제안된 advanced 언어모델 기술들은 그들의 간단한 베이스라인에서 아주 극소량의 성능개선만이 있었고, 실제로는(현실적으로) 아주 드물게 사용된다.

- 기존 연구 :
- 통계 언어 모델링에서 '고정 길이 컨텍스트'를 가진 피드포워드 신경망을 사용한 인공 신경망 - Bengio
- 이 접근 방식은 예외적으로 성공적이었으며 이 단일 모델이 클래스 기반 모델을 포함한 여러 다른 모델의 혼합보다 성능이 우수했다.
- 신경망 기반 모델이 기존에 비해 몇몇 task들에서 음성 인식을 크게 향상시킨다는 것을 보여주었다.
- 문제점 :
- 피드포워드 네트워크가 고정 길이 컨텍스트를 사용해야 한다는 것이다.
- 신경망이 다음 단어를 예측할 때 5개에서 10개의 선행 단어만 본다는 것을 의미한다
- 인간은 더 큰 맥락을 활용한다.
- 따라서, '임의의 길이를 가진 컨텍스트'에 대해 '시간 정보를 암시적으로 인코딩'하는 모델을 만들고자 한다.

2. Model description

그림 5-25 RNNLM의 신경망(왼쪽이 펼치기 전, 오른쪽은 펼친 후)

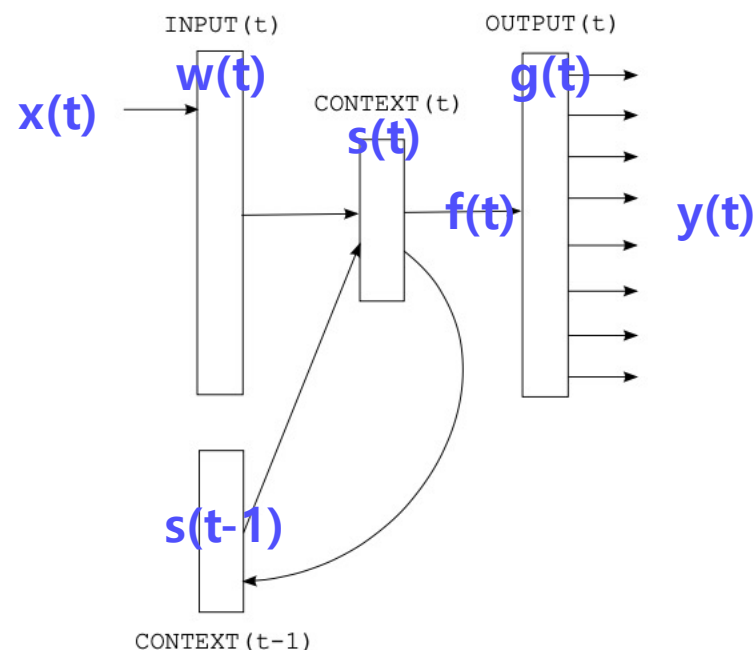


2. Model description

- RNN은 제한된 크기의 컨텍스트를 사용하지 않는다. 반복적인 연결을 사용함으로써, 정보는 이러한 네트워크 내에서 임의적으로 오랜 시간 동안 순환할 수 있다.
- SGD에 의한 장기 의존성 학습이 상당히 어려울 수 있다는 주장도 종종 제기되기도 한다.
- 순환 신경망의 가장 간단한 버전을 소개할 것이고, 구현하고 훈련하는 것이 매우 쉽다.
- input layer : x
- hidden layer : s (컨텍스트 계층 or state)
- output layer: y
- 시간 : t
- $x(t)$ 는 시간 t 일때의 x 를 의미하고, $y(t)$, $s(t)$ 도 마찬가지
- 입력 벡터 $x(t)$ 는 w (현재 단어의 분산 표현)와 시간 $t - 1$ 에서의 s (context layer)연결하여 형성된다.

2. Model description

- 현재 단어와 이전 스텝의 hidden layer가 input



$$x(t) = w(t) + s(t-1)$$

$$s_j(t) = f \left(\sum_i x_i(t) u_{ji} \right)$$

$$y_k(t) = g \left(\sum_j s_j(t) v_{kj} \right)$$

where $f(z)$ is sigmoid activation function:

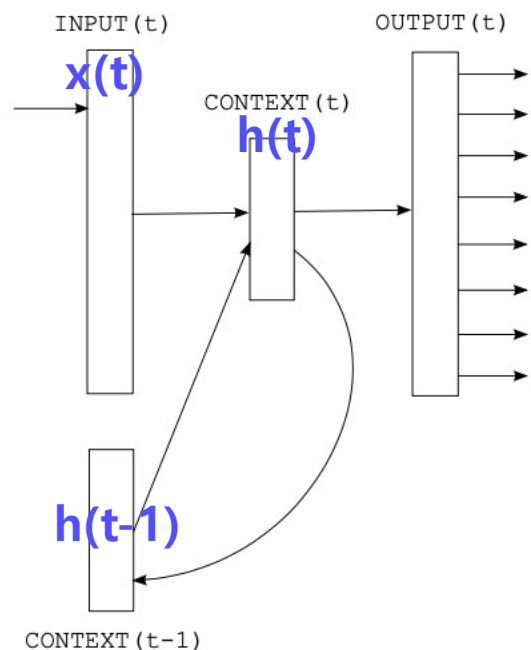
$$f(z) = \frac{1}{1 + e^{-z}}$$

and $g(z)$ is softmax function:

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

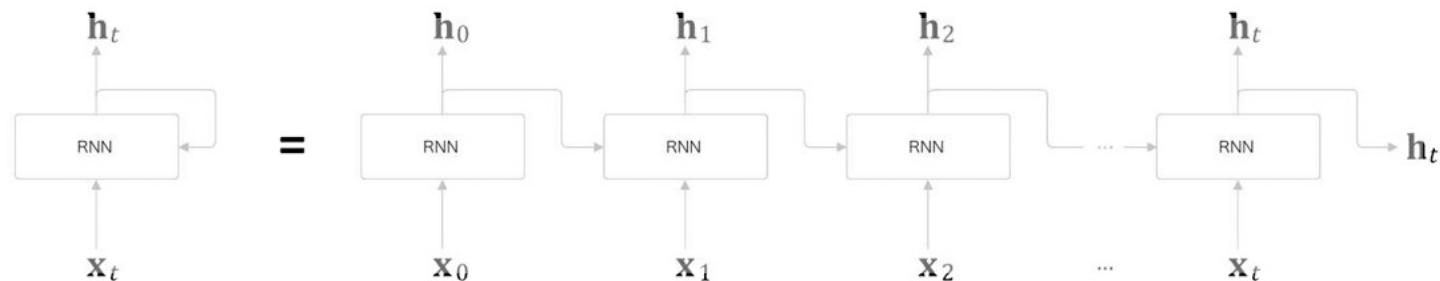
2. Model description

- 현재 단어와 이전 스텝의 hidden layer가 input



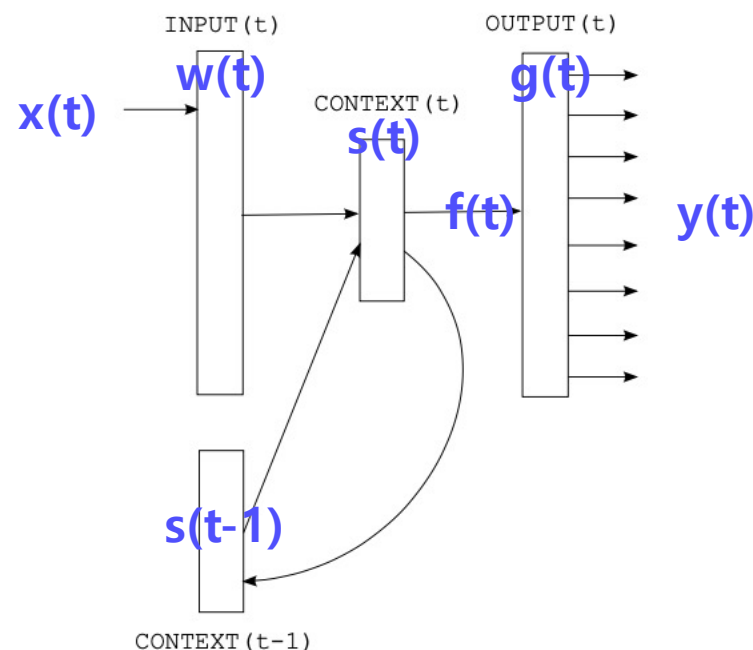
$$\mathbf{h}_t = \tanh(\mathbf{h}_{t-1} \mathbf{W}_h + \mathbf{x}_t \mathbf{W}_x + \mathbf{b})$$

그림 5-8 RNN 계층의 순환 구조 펼치기



2. Model description

- 현재 단어와 이전 스텝의 hidden layer가 input



$$x(t) = w(t) + s(t-1)$$

$$s_j(t) = f \left(\sum_i x_i(t) u_{ji} \right)$$

$$y_k(t) = g \left(\sum_j s_j(t) v_{kj} \right)$$

where $f(z)$ is sigmoid activation function:

$$f(z) = \frac{1}{1 + e^{-z}}$$

and $g(z)$ is softmax function:

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$


2. Model description

- 입력 벡터 $x(t)$ 는 1-of-N 코딩을 사용하여 시간 t 에서의 단어를 나타냅니다.
ex) $x(t_1)=(1,0,0,0)$ $x(t_2)=(0,1,0,0)$ $x(t_3)=(0,0,1,0)$ $x(t_4)=(0,0,0,1)$ -> $N = 4$
- 벡터 x 의 크기는 vocab_size V (실제로는 30000 - 200000)와 같다.
- context layer s 의 크기는 일반적으로 30 - 500개.
- hidden layer의 크기는 훈련 데이터의 양을 반영해야 한다
 - 많은 양의 데이터의 경우, 큰 hidden layer가 필요하다.
- 네트워크는 훈련 말뭉치의 모든 데이터가 순차적으로 몇개의 에폭에 의해 훈련된다.
- 가중치는 작은 값(평균이 0이고 분산이 0.1인 랜덤 가우스 노이즈)으로 초기화됩니다.
- SGD와 함께 기본 역전파 알고리즘을 사용한다.
- $lr = 0.1$
- validation data의 log-likelihood가 증가하도록 훈련
- 유의한 개선이 관찰되지 않으면 학습률을 각 에폭이 시작될 때 절반으로 감소한다.
- 큰 개선이 없으면 훈련이 종료됩니다. 수렴은 보통 10-20 에폭 이후에 이루어진다.

2. Model description

- 실험 과정에서 가중치에 불이익을 주는 방식의 정규화는 큰 효과가 없었다.
- output layer $y(t)$ 는 이전 단어 $w(t)$ 와 $s(t - 1)$ 가 주어진 다음 단어의 확률 분포를 나타냅니다.
- Softmax는 이 확률 분포가 유효한지 확인합니다.
- 모든 단어 m 에 대해 $y_m(t) > 0$, $\sum_k y_k(t) = 1$.
- loss는 cross entropy error에 따라 계산되고 weight update는 기본 역전파 알고리즘으로 업데이트된다.

$$\text{error}(t) = \text{desired}(t) - y(t)$$

 target

- 여기서 desired는 특정 컨텍스트에서 예측되어야 하는 단어를 나타내는 1-of-N 코딩을 사용하는 벡터이고 $y(t)$ 는 네트워크로부터의 실제 출력이다.

2. Model description

- 통계 언어 모델링의 train 단계와 test 단계는 일반적으로 시험 데이터가 처리될 때 모델이 업데이트되지 않는다는 점에서 다르다.
- 그래서 만약 test set에 새로운 사람 이름이 반복적으로 나오더라도, 그 이름은 아주 적은 확률을 얻게 될 것이다.
- 장기 메모리는 시냅스 자체에 위치해야 하며, 네트워크는 테스트 단계 중에도 훈련을 계속해야 한다고 가정할 수 있다. 이러한 모델을 동적 모델이라고 합니다.
- train 단계에서는 모든 데이터가 에폭들에서 여러 번 네트워크에 제시되지만 동적 모델은 테스트 데이터를 처리할 때 한 번만 업데이트된다.
- 이것은 물론 최적의 해결책은 아니지만, 정적 모델에 대한 큰 복잡성 감소를 얻기에 충분하다.
- 신경망이 연속적인 공간에서 학습한다는 점은 백오프 모델의 캐시 기술과 매우 유사하므로, 'dog'와 'cat'이 관련된 경우 테스트 데이터에서 'dog'가 자주 발생하는 것도 'cat'의 확률을 증가시킨다.

- 이렇게 동적으로 업데이트된 모델은 새로운 도메인에 자동으로 적응할 수 있다.
- 그러나 음성 인식 실험에서 history는 recognizer에 의해 주어진 가설로 표현되며 인식 오류를 포함한다.
 - 이는 캐시 n그램 모델의 성능이 떨어진다.
- 이번 training은 시간 경과에 따른 잘린 역전파를 사용한다.
- 현재 시간 단계에 대해서만 계산된 loss로 가중치가 업데이트되므로 최적은 아니다. 이걸 해결하기 위해 시간을 통한 역전파(BPTT) 알고리즘이 사용된다.
- 피드포워드 신경망과 recurrent 신경망 사이의 주요 차이점 중 하나는 훈련 전에 조정하거나 임시로 선택해야 하는 매개 변수의 양이다.
- RNN LM의 경우 hidden(context) layer의 크기만 선택하면 된다.
- 피드포워드 네트워크의 경우, 단어를 저차원 공간에 투영하는 레이어의 크기, hidden_layer의 크기 및 context length 등을 조정해야 한다.

- 성능을 향상시키기 위해, 우리는 (train text에서) 임계값보다 덜 자주 발생하는 모든 단어를 특수 rare 토큰으로 병합한다. 단어 확률은 다음과 같이 계산된다.

$$P(w_i(t+1)|w(t), s(t-1)) = \begin{cases} \frac{y_{rare}(t)}{C_{rare}} & \text{if } w_i(t+1) \text{ is rare,} \\ y_i(t) & \text{otherwise} \end{cases} \quad (7)$$

- 여기서 C_{rare} 는 임계값보다 덜 자주 발생하는 어휘의 단어 수이다.

3 WSJ experiments

- RNNLM의 성능을 평가하기 위해 몇 가지 표준 음성 인식 작업을 선택했다.
- 먼저 DARPA WSJ'92 및 WSJ'93 데이터 세트에서 100개의 베스트 목록을 복원한 후 결과를 보고한다.
- Oracle WER은 devset의 경우 6.1%, evalset의 경우 9.5%입니다. 언어 모델에 대한 훈련 데이터는 Xu[8]에서 사용한 것과 동일합니다.

Table 2: *Comparison of various configurations of RNN LMs and combinations with backoff models while using 6.4M words in training data (WSJ DEV).*

Model	PPL		WER	
	RNN	RNN+KN	RNN	RNN+KN
KN5 - baseline	-	221	-	13.5
RNN 60/20	229	186	13.2	12.6
RNN 90/10	202	173	12.8	12.2
RNN 250/5	173	155	12.3	11.7
RNN 250/2	176	156	12.0	11.9
RNN 400/10	171	152	12.5	12.1
3xRNN static	151	143	11.6	11.3
3xRNN dynamic	128	121	11.3	11.1

Table 3: *Comparison of WSJ results obtained with various models. Note that RNN models are trained just on 6.4M words.*

Model	DEV WER	EVAL WER
Lattice 1 best	12.9	18.4
Baseline - KN5 (37M)	12.2	17.2
Discriminative LM [8] (37M)	11.5	16.9
Joint LM [9] (70M)	-	16.7
Static 3xRNN + KN5 (37M)	11.0	15.5
Dynamic 3xRNN + KN5 (37M)	10.7	16.3 ⁴

- 이전 실험은 공정한 기준선에 비해 매우 흥미로운 개선을 보여주지만, 유효한 비판은 그러한 실험에 사용된 음향 모델이 최첨단과는 거리가 멀다는 것이고, 아마도 그러한 경우에 개선을 얻는 것이 잘 조정된 시스템을 개선하는 것보다 더 쉬울 것이다. 더욱 중요한 것은 기준 백오프 모델을 훈련하는 데 사용되는 37M 또는 70M 단어가 작업에 가능한 단어보다 훨씬 적다는 사실이다. 최첨단 시스템 상태에서 의미 있는 개선을 얻을 수 있다는 것을 보여주기 위해, 우리는 NIST RT05 평가에 사용된 AMI 시스템에서 생성된 격자를 실험했다[13]. 테스트 데이터 세트는 독립 헤드셋 조건에 대한 NIST RT05 평가였다.

Table 4: Comparison of very large back-off LMs and RNN LMs trained only on limited in-domain data (5.4M words).

Model	WER static	WER dynamic
RT05 LM	24.5	-
RT09 LM - baseline	24.1	-
KN5 in-domain	25.7	-
RNN 500/10 in-domain	24.2	24.1
RNN 500/10 + RT09 LM	23.3	23.2
RNN 800/10 in-domain	24.3	23.8
RNN 800/10 + RT09 LM	23.4	23.1
RNN 1000/5 in-domain	24.2	23.7
RNN 1000/5 + RT09 LM	23.4	22.9
3xRNN + RT09 LM	23.3	22.8

- 반복 신경망은 우리의 모든 실험에서 최첨단 백오프 모델을 크게 능가했다. 특히 백오프 모델이 RNN LM보다 훨씬 더 많은 데이터에 대해 훈련되었을 때조차도 말이다.
- WSJ 실험에서, 단어 오류율 감소는 동일한 양의 데이터에 대해 훈련된 모델의 경우 약 18%, RNN 모델보다 5배나 많은 데이터를 가지고 있던 백오프 모델이 훈련되었을 때 약 12%였다.
- NIST RT05의 경우 모델이 5.4M개만으로 훈련되었다고 결론을 내릴 수 있다.도메인 내 데이터의 워드는 수백 배 더 많은 데이터에 대해 훈련된 빅 백오프 모델을 능가할 수 있다. 얻어진 결과는 언어 모델링이 n-gram을 세는 것뿐이며, 결과를 개선할 수 있는 유일한 합리적인 방법은 새로운 훈련 데이터를 얻는 것이라는 신화를 깨뜨리고 있다.

- 표 2에 보고된 복잡성 개선은 유사한 데이터 세트에 대해 보고된 것 중 가장 큰 것 중 하나로, 동적 학습(본 논문에서는 동적 모델이라고도 하며, 비지도 LM 훈련 기술과 매우 유사한 음성 인식 맥락에서)에 매우 중요한 영향을 미친다.
- WER은 약간의 영향을 받고 테스트 데이터의 올바른 순서를 요구하는 반면, 동적 학습은 캐시와 같은 정보와 트리거와 같은 정보를 얻는 방법을 자연스럽게 제공하기 때문에 더 자세히 조사되어야 한다.
- 만약 우리가 정말로 언어를 배울 수 있는 모델을 만들고 싶다면, 동적 학습은 매우 중요하다.
RNN을 학습하기 위한 BPTT에 대한 추가 연구가 있다면 더 좋을 것이다.
- 캐시 모델은 BPTT로 훈련된 동적 모델에도 여전히 보완적인 정보를 제공하기 때문에 단순한 반복 신경망이 진정으로 긴 컨텍스트 정보를 캡처할 수 있는 것 같지는 않다.

- Task에서 task나 언어 관련 전제가 없었기 때문에 기계 번역이나 OCR과 같은 백오프 언어 모델을 사용하는 모든 종류의 애플리케이션에서 RNN 기반 모델을 거의 쉽게 사용할 수 있다.
- 특히 굴절 언어(프랑스어 같은 애들) 또는 어휘가 큰 언어를 포함하는 작업에서 성능이 좋았다. NN 기반 모델이 이미 이런 점에서 좋다는 연구가 있었고 RNN도 이것 포함함.
- 우리의 연구에서 보고된 매우 좋은 결과 외에도 제안된 RNNLM 언어 모델링을 기계 학습, 데이터 압축 및 인지 과학 연구와 더 밀접하게 연결하기 때문에 흥미롭다. 우리는 이러한 연결들이 앞으로 더 잘 이해되기를 바랍니다.