

# Adversarial examples for evaluating reading comprehension systems

Robin Jia, Percy Liang

집현전 중급반 12조

김주찬, 박병학(발표자), 이창호

# Index

---

01

## Introduction

- Purpose
- Baseline

02

## Generation

- AddSent
- AddOneSent
- AddAny
- AddCommon

03

## Experiment

- Performance
- Categorize
- Transferability
- Fine-Tuning

04

## Conclusion



---

01

## Introduction

- Purpose
- Baseline

02

## Generation

- AddSent
- AddOneSent
- AddAny
- AddCommon

03

## Experiment

- Performance
- Categorize
- Transferability
- Fine-Tuning

04

## Conclusion



# Purpose

- 많은 Reading Comprehension System들의 성능은 빠르게 향상되고 있지만, 이 모델들이 실제로 언어를 이해를 하고 있는 것인지는 확실치 않음
- 이를 확인하기 위해 문단에 Adversarial Sentences가 추가되었을 때에도 질문에 대한 답변을 잘 하는지를 확인하기 위한 데이터셋을 생성

**Article:** Super Bowl 50

**Paragraph:** “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”

**Question:** “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”

**Original Prediction:** John Elway

**Prediction under adversary:** Jeff Dean

# Baseline

## - Task

- Stanford Question Answering Dataset (SQuAD)
  - 위키피디아 글에 대해 사람이 생성한 107,785개의 RC Question으로 구성된 데이터셋
  - 데이터셋 하나마다 Paragraph + Question + Answer로 구성

## - Models

- 정답에 대한 확률 분포를 예측하는 모델들을 사용
  - BiDAF(Bidirectional Attention Flow)
  - Match-LSTM
- Validation에는 추가적으로 12개의 모델을 사용

**Article:** Super Bowl 50

**Paragraph:** "Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. *Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"

**Question:** "What is the name of the quarterback who was 38 in Super Bowl XXXIII?"

**Original Prediction:** John Elway

**Prediction under adversary:** Jeff Dean

# Baseline

## - Evaluation

- F1 Score를 활용
- 실제 정답과 예측된 정답 사이의 F1 score의 평균을 사용
- Adversarial Evaluation에서는 adversary가 추가된 p에 대해 예측된 정답과 실제 정답 사이의 F1 score를 구해 adversarial accuracy를 구함

$$\text{Acc}(f) \stackrel{\text{def}}{=} \frac{1}{|D_{\text{test}}|} \sum_{(p,q,a) \in D_{\text{test}}} v((p,q,a), f),$$

$$\text{Adv}(f) \stackrel{\text{def}}{=} \frac{1}{|D_{\text{test}}|} \sum_{(p,q,a) \in D_{\text{test}}} v(A(p,q,a), f).$$



---

01

## Introduction

- Purpose
- Baseline

02

## Generation

- AddSent
- AddOneSent
- AddAny
- AddCommon

03

## Experiment

- Performance
- Categorize
- Transferability
- Fine-Tuning

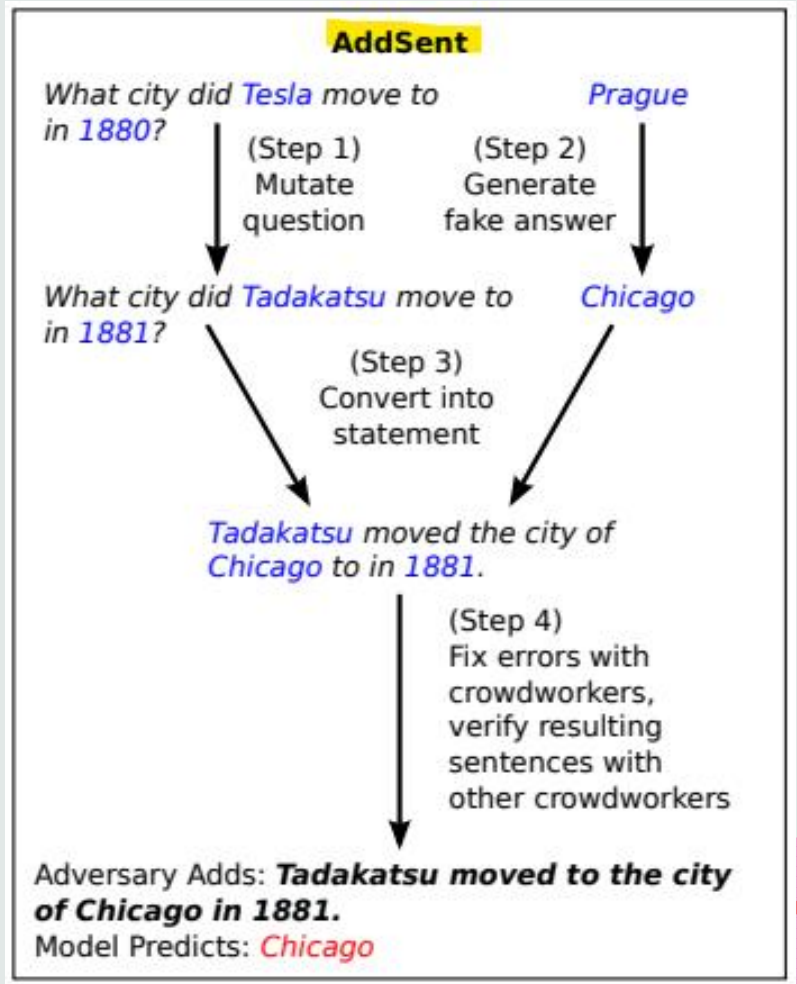
04

## Conclusion



# AddSent

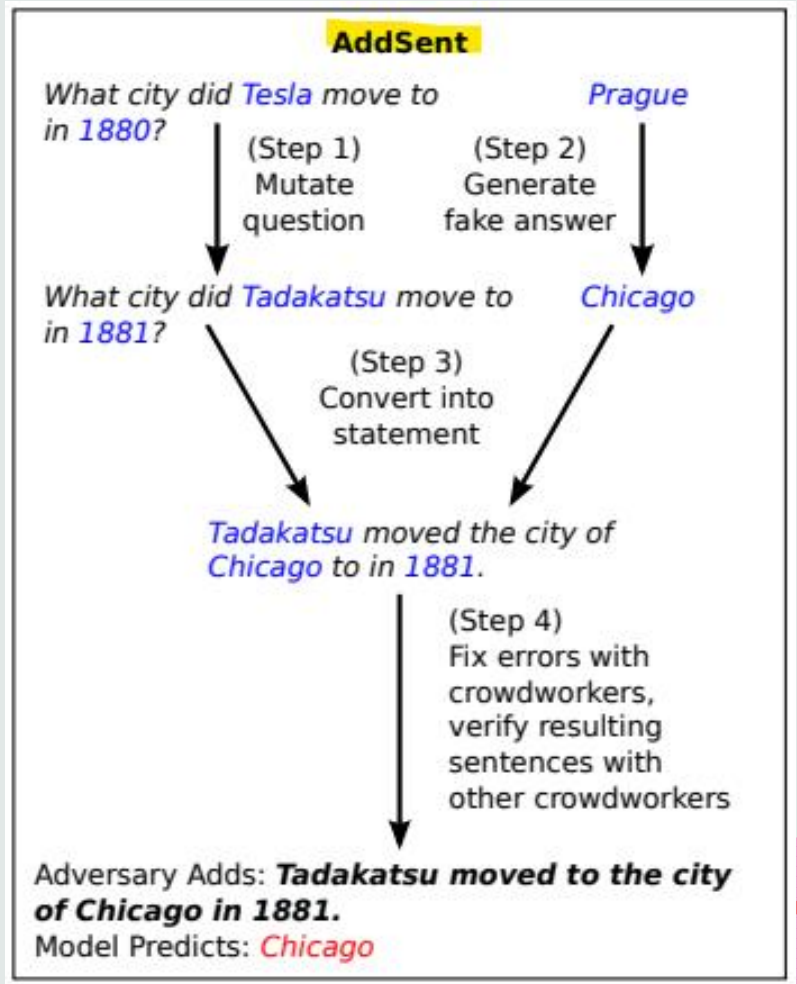
- 4가지 단계를 거쳐 질문과 비슷한 문장을 생성하여 문단의 제일 끝에 붙임
- 1. 명사와 형용사를 WordNet에서 얻은 반의어로 바꾸고, Named Entities를 GloVe 단어벡터공간에서 가장 유사한 단어로 교체
- 2. 원래 정답과 같은 "종류"의 정답 생성
- 3. 1, 2번 과정을 거쳐 얻은 가짜 질문과 가짜 정답을 평서문으로 전환
- 4. Crowdsourcing을 통한 문장의 오류 수정





# AddOneSent

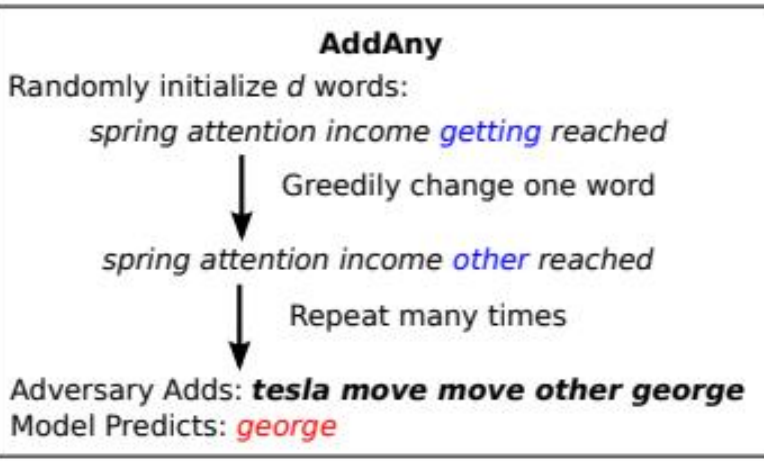
- AddSent의 경우에는 model이 최악의 정답을 내놓은 경우를 선택한다는 점에서 어느정도 model dependent
- Model independent한 case에 대한 가능성을 위해, AddOneSent 도입
- AddOneSent는 AddSent에서 마지막 최악의 정답을 내놓는 경우를 선택하는 것이 아닌, 랜덤으로 하나를 선택하여 사용



# AddAny

- 문법에 상관없이 d개의 단어로 이루어진 sequence를 생성
- 1. Random Sampled common words와 Words in Question으로 구성된 단어들의 seed set을 구성
- 2. Iteration을 돌며 d의 크기 만큼 greedy하게 단어를 바꿈
- 3. 각 iteration에 대해 모델의 output distribution에 대한 F1-score를 최소화 하는 경우를 선택

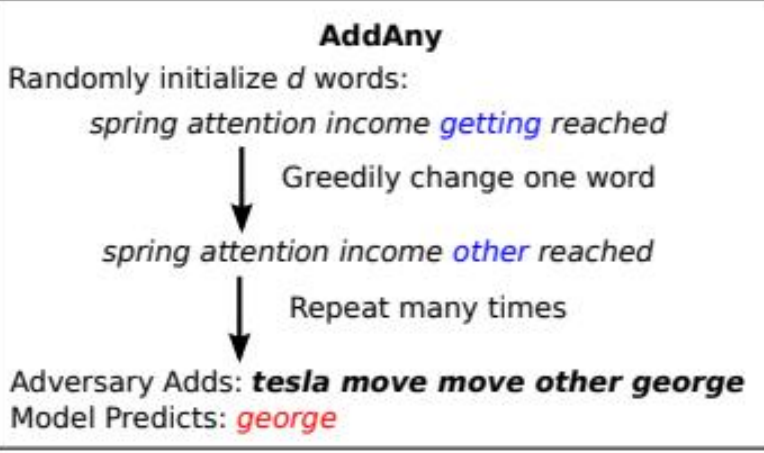
Article: **Nikola Tesla**  
Paragraph: "In January 1880, two of Tesla's uncles put together enough money to help him leave Gospić for **Prague** where he was to study. Unfortunately, he arrived too late to enroll at Charles-Ferdinand University; he never studied Greek, a required subject; and he was illiterate in Czech, another required subject. Tesla did, however, attend lectures at the university, although, as an auditor, he did not receive grades for the courses."  
Question: "What city did Tesla move to in 1880?"  
Answer: **Prague**  
Model Predicts: **Prague**



# AddCommon

- AddAny의 경우는 생성된 문장에 질문들의 단어가 포함되어 있다.
- 이를 피하기 위해 단어의 seed set을 생성시에 random sampled common words만 포함하여 문장을 생성

Article: **Nikola Tesla**  
Paragraph: "In January 1880, two of Tesla's uncles put together enough money to help him leave Gospić for **Prague** where he was to study. Unfortunately, he arrived too late to enroll at Charles-Ferdinand University; he never studied Greek, a required subject; and he was illiterate in Czech, another required subject. Tesla did, however, attend lectures at the university, although, as an auditor, he did not receive grades for the courses."  
Question: "What city did Tesla move to in 1880?"  
Answer: **Prague**  
Model Predicts: **Prague**





---

01

## Introduction

- Purpose
- Baseline

02

## Generation

- AddSent
- AddOneSent
- AddAny
- AddCommon

03

## Experiment

- Performance
- Categorize
- Transferability
- Fine-Tuning

04

## Conclusion



# Performance

- 어떤 모델도 adversary에 대처를 잘하는 모습을 보이지 않음
- AddAny가 AddSent보다 더 효과적으로 성능을 떨어뜨림

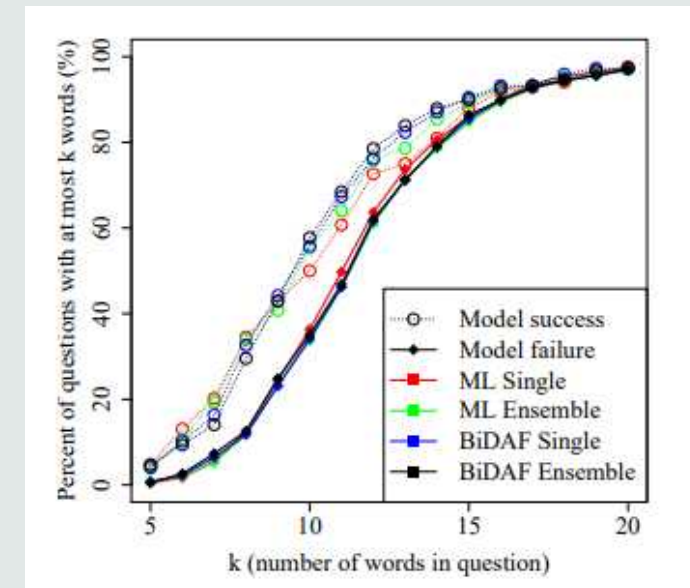
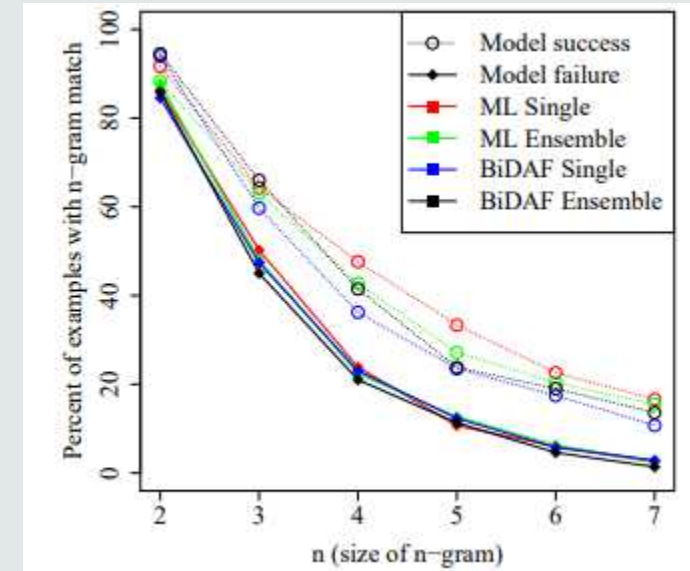
	Match Single	Match Ens.	BiDAF Single	BiDAF Ens.
Original	71.4	75.4	75.5	80.0
ADDSENT	27.3	29.4	34.3	34.2
ADDONESENT	39.0	41.8	45.7	46.9
ADDANY	7.6	11.7	4.8	2.7
ADDCOMMON	38.9	51.0	41.7	52.6

Model	Original	ADDSENT	ADDONESENT
ReasoNet-E	<b>81.1</b>	39.4	49.8
SEDT-E	80.1	35.0	46.5
BiDAF-E	80.0	34.2	46.9
Mnemonic-E	79.1	<b>46.2</b>	<b>55.3</b>
Ruminating	78.8	37.4	47.7
jNet	78.6	37.9	47.0
Mnemonic-S	78.5	<b>46.6</b>	<b>56.0</b>
ReasoNet-S	78.2	39.4	50.3
MPCM-S	77.0	40.3	50.0
SEDT-S	76.9	33.9	44.8
RaSOR	76.2	39.5	49.5
BiDAF-S	75.5	34.3	45.7
Match-E	75.4	29.4	41.8
Match-S	71.4	27.3	39.0
DCR	69.3	37.8	45.1
Logistic	50.4	23.2	30.4

# Performance

- 모델이 adversary가 있음에도 잘 예측을 한 경우(Model Success)는 2가지가 존재

1. 질문이 원본 문장과 정확한 n-gram 일치가 있는 경우
2. 짧은 질문인 경우



# Categorize AddSent

- AddSent의 Model Failure의 경우에, 96.6%는 adversary의 span을 정답으로 예측함.
- 100개의 BiDAF Ensemble Failure Sample을 추출했을 때
  - 75개는 Entity name의 변경
  - 17개는 숫자나 날짜의 변경
  - 33개는 질문의 단어의 반의어 사용

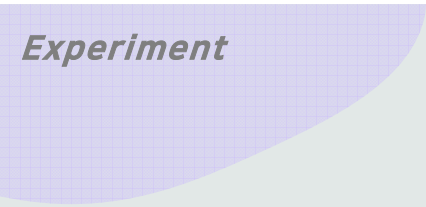


# Transferability

- 특정 모델에 맞춰 adversary example을 만들었을 때, 이 example들이 다른 모델에서도 효과가 있을 것인가?
- 실험 결과 어떤 모델에 효과가 있는 example의 경우, 다른 모델에서도 효과가 있음을 알 수 있다.
- 특히 AddSent의 경우에 전반적으로 효과적인 모습을 보임

Targeted Model	Model under Evaluation			
	ML Single	ML Ens.	BiDAF Single	BiDAF Ens.
<b>ADDSENT</b>				
ML Single	27.3	33.4	40.3	39.1
ML Ens.	31.6	29.4	40.2	38.7
BiDAF Single	32.7	34.8	34.3	37.4
BiDAF Ens.	32.7	34.2	38.3	34.2
<b>ADDANY</b>				
ML Single	7.6	54.1	57.1	60.9
ML Ens.	44.9	11.7	50.4	54.8
BiDAF Single	58.4	60.5	4.8	46.4
BiDAF Ens.	48.8	51.1	25.0	2.7

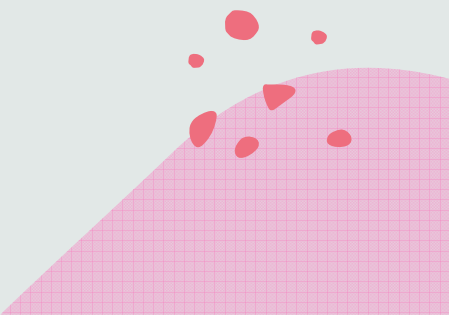




# Fine-Tuning

- Adversarial Example에 대해 모델을 fine-tuning시키면 어떨 것인가에 대한 실험
- AddSentMod: 문단의 끝이 아닌 도입부에 생성한 문장을 추가 + 가짜 정답의 다른 set을 이용
- AddSent에 대해서는 fine-tuning이 효과가 있었지만, AddSentMod에서는 실패
- 이를 통해 AddSent에서는 마지막 문장을 고려하지 않는다는 식의 overfitting이 발생하였음을 확인할 수 있음

Test data	Training data	
	Original	Augmented
Original	75.8	75.1
ADDSENT	34.8	70.4
ADDSENTMOD	34.3	39.2





---

01

## Introduction

- Purpose
- Baseline

02

## Generation

- AddSent
- AddOneSent
- AddAny
- AddCommon

03

## Experiment

- Performance
- Categorize
- Transferability
- Fine-Tuning

04


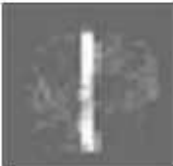
## Conclusion



# Conclusion

- 표준 평가 지표에 따른 모델의 성능은 성공적이었음에도 불구하고, 기존의 Reading Comprehension System은 adversarial evaluation에서 성능이 떨어짐
- 본 논문에서 실시한 adversarial evaluation은 기존 모델이 지나치게 안정적이라는 것을 보여줌
- 이러한 adversarial evaluation을 최적화 하려면 새로운 전략이 필요함
- 언어를 진정으로 이해하는 시스템 구축을 위해 모든 SQuAD 시스템에서 AddSent를 실행하는 스크립트와 AddAny코드를 릴리즈 하였음
- 해당 작업이 언어를 더 깊이 이해하는 보다 정교한 모델의 개발에 동기를 부여하기를 바람

# Conclusion

	Image Classification	Reading Comprehension
Possible Input		Tesla moved to the city of Chicago in 1880.
Similar Input		Tadakatsu moved to the city of Chicago in 1881.
Semantics	Same	Different
Model's Mistake	Considers the two to be different	Considers the two to be the same
Model Weakness	Overly sensitive	Overly stable

# Conclusion

- 표준 평가 지표에 따른 모델의 성능은 성공적이었음에도 불구하고, 기존의 Reading Comprehension System은 adversarial evaluation에서 성능이 떨어짐
- 본 논문에서 실시한 adversarial evaluation은 기존 모델이 지나치게 안정적이라는 것을 보여줌
- 이러한 adversarial evaluation을 최적화 하려면 새로운 전략이 필요함
- 언어를 진정으로 이해하는 시스템 구축을 위해 모든 SQuAD 시스템에서 AddSent를 실행하는 스크립트와 AddAny코드를 릴리즈 하였음
- 해당 작업이 언어를 더 깊이 이해하는 보다 정교한 모델의 개발에 동기를 부여하기를 바람