EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks

Jason Wei, Kai Zou

2019.08.25

발표자 : 김산

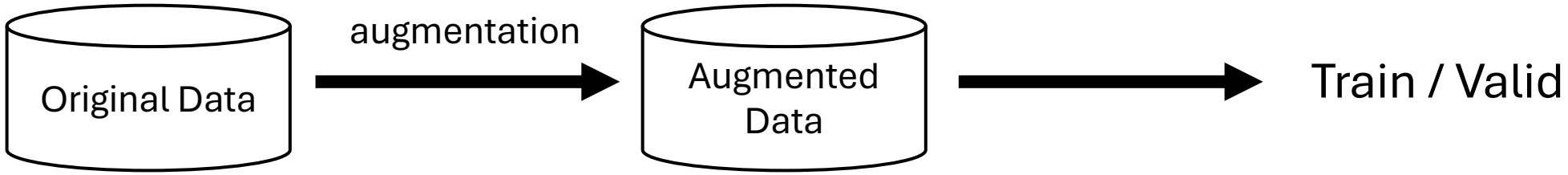
Introduction

Data Augmentation?

데이터 증강 → 원본 데이터의 label을 보존하면서 새로운 데이터를 생성

Why Data Augmentation?

1. Overfitting 방지
2. 외부에서 데이터를 수집 및 분류 하는데 많은 비용이 필요.

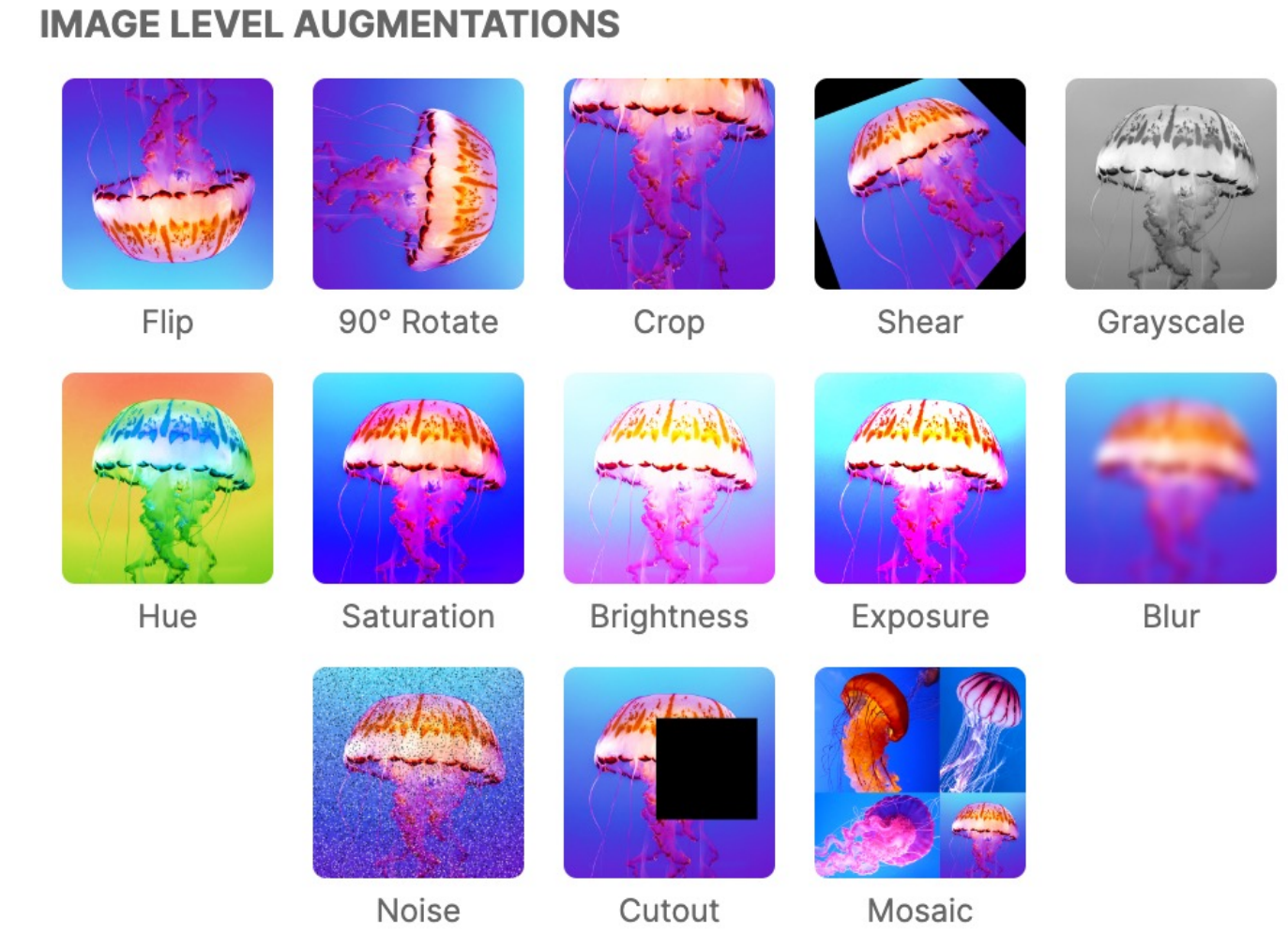


Introduction

Image Augmentation

Image는 변화를 줘도 본래 label을 보존한다.

때문에 다양한 augmentation 기법이 가능



Introduction

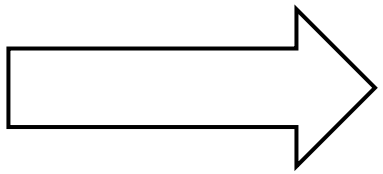
Text Augmentation

단어가 바뀌면 의미가 변하기 때문에 Image 증강에 비해 어려움

문서의 label을 보존하면서 표현을 다양화 하는 것이 목표

- 문장이 가지는 의미가 있기 때문에 작은 변화로 다른 의미가 될 수 있다.
- 비슷한 문장으로 바꾸는 것에 대한 평가가 어렵다.

This is dog



Is this dog

It is cat

Introduction

Randon Noise Injection

간단한 텍스트 편집기법을 이용하여 데이터를 증강시키는 방법

Back Translation

번역기를 이용해 원본과 비슷한 텍스트 생성

Generative Methods

Pre-trained model을 이용해 데이터를 생성

EDA

Synonym Replacement (SR)	문장 내 불용어가 아닌 n개의 단어를 선택해 유의어로 교체
Random Insertion (RI)	문장 내 불용어가 아닌 단어 중 임의의 단어를 임의의 자리에 삽입한다. (n번 수행)
Random Swap(RS)	문장 내 임의의 두 단어를 선택해 위치를 바꾼다. (n번 수행)
Random Deletion (RD)	문장 내 임의의 단어를 p확률로 제거한다.

SR은 이전에도 사용했지만, RI, RS, RD는 이전에 연구되지 않았다.

EDA

원문 : 나는 프로그래밍하는 것을 좋아한다.

SR : 나는 게임하는 것을 좋아한다.

RI : 나는 프로그래밍 하는 것을 지금 좋아한다.

RS : 프로그래밍하는 것을 나는 좋아한다.

RD : 프로그래밍하는 것을 좋아한다.

긴 문장은 짧은 문장에 비해 단어가 더 많아,

원래 label을 보존하며 상대적으로 noise의 영향을 덜 받는다.

Experimentent

성능 평가를 위한 5개의 benchmark test classification task

1. SST-2 : Stanford Sentiment Treebank
2. CR : Customer Review
3. SUBJ : Subjectively/Objectively dataset
4. TREC : Question type dataset
5. PC : Pro-Con dataset

2개의 network architeture

1. LSTM-RNN
2. CNN

Experimentent

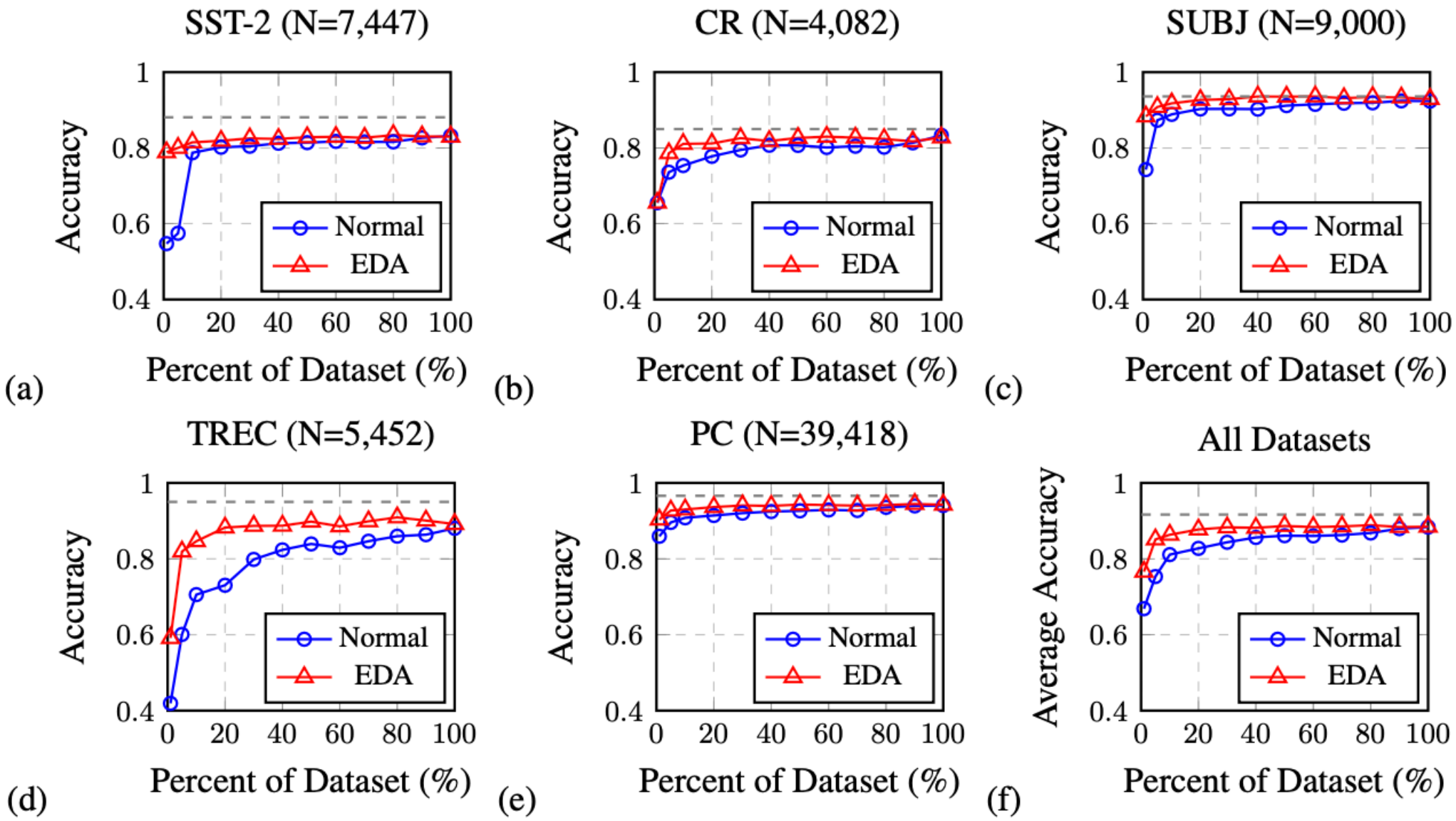
Model	Training Set Size			
	500	2,000	5,000	full set
RNN	75.3	83.7	86.1	87.4
+EDA	79.1	84.4	87.3	88.3
CNN	78.6	85.6	87.7	88.3
+EDA	80.7	86.4	88.3	88.8
Average	76.9	84.6	86.9	87.8
+EDA	79.9	85.4	87.8	**88.6**

Dataset 크기별 성능을 비교

500 set의 경우 3.0% 향상

Full set의 경우 0.8% 향상

Experimentent

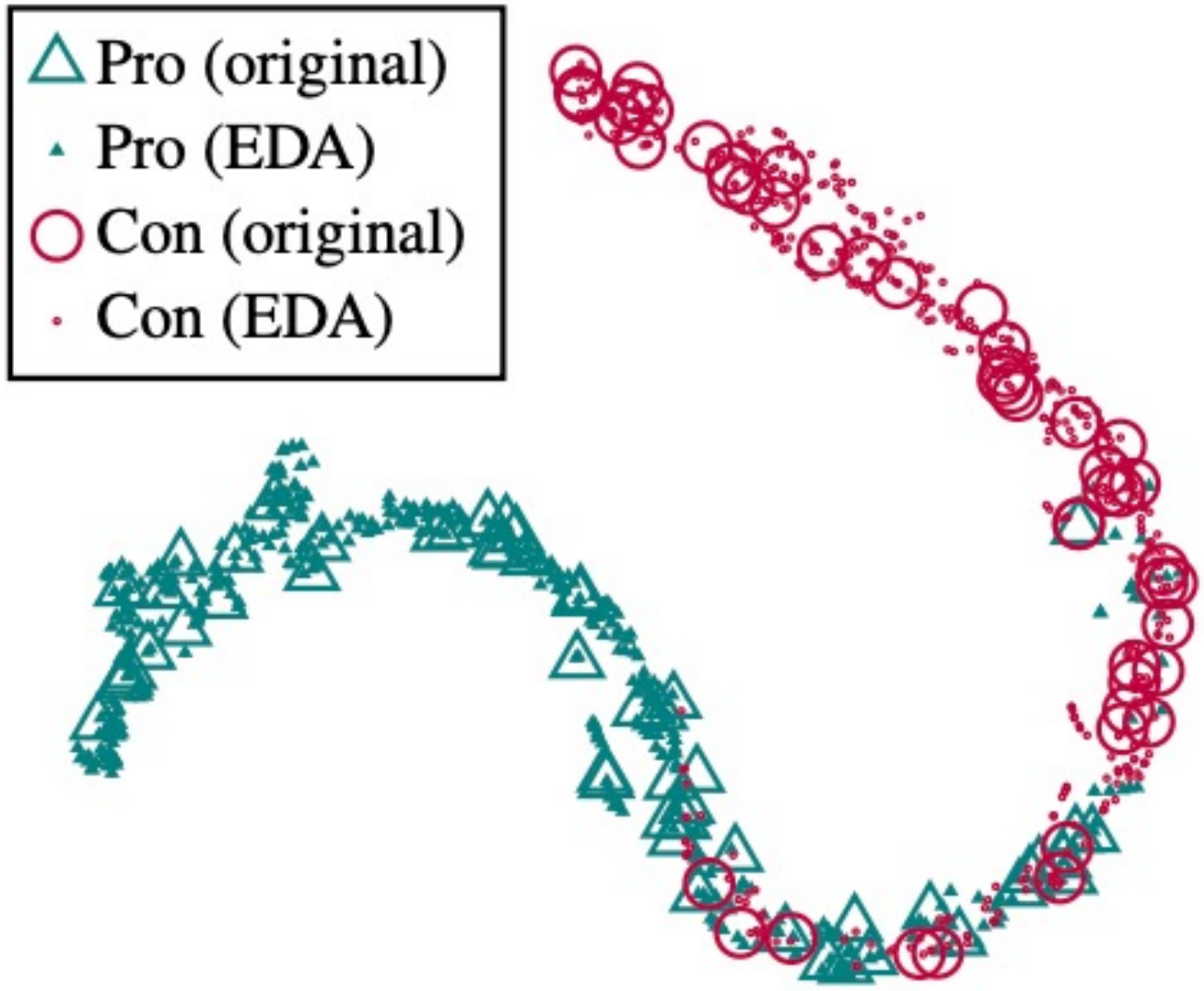


Experimentent

Pro-Con classification

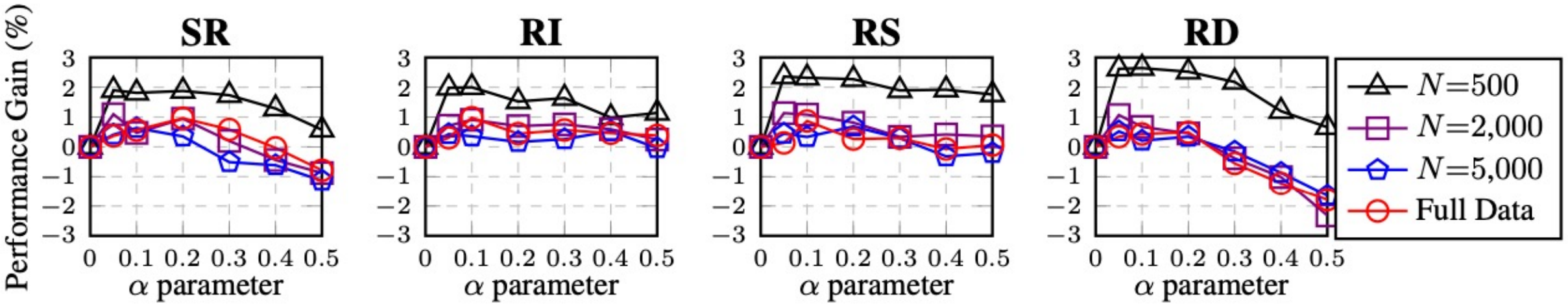
문장 당 9개의 문장 생성

→ 증강된 문장이 원래 문장의 주변에 분포한 것을 확인



Experimentent

- EDA별 성능 확인



α가 크면 문장의 identity가 변함

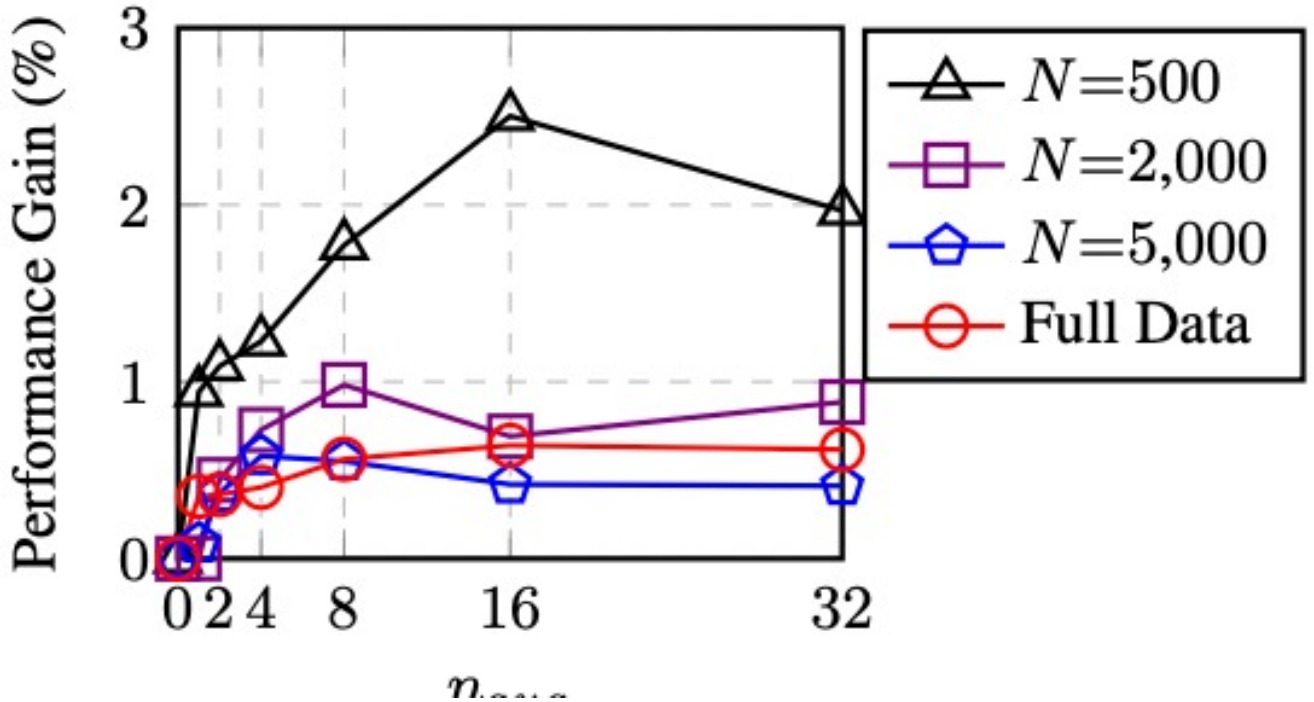
순서가 연산에서 유지

많은 swap은 문장의 순서를 바꾼 것과 같음

많은 삭제는 문장의 이해를 어렵게 함

Experimentent

- 문장 당 증강 개수별 성능



N_{train}	α	n_{aug}
500	0.05	16
2,000	0.05	8
5,000	0.1	4
More	0.1	4

Conclusion

1. SR, RI, RD, RS를 사용하여 데이터 증강
2. 소규모 데이터에서 성능을 향상시키고 과적합의 가능성을 낮춘다.
3. 적절한 노이즈 생성으로 데이터 부족으로 발생하는 과적합을 방지.