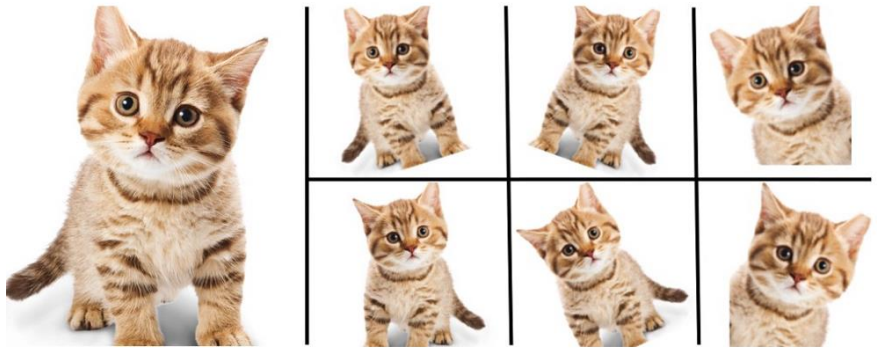
EDA : Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks

김웅희

- Author
  - Jason Wei
  - Kai Zou
- Title of Confenrence
  - EMNLP-IJCNLP 2019

• Introduction

- 이 논문에서 제시하는 문제
  - 기계 학습은 데이터양이 많을수록 정확도가 높아짐
  - 기존의 Data Augmentation은 컴퓨터 비전과 음성처리에만 치중되어 있음
    - 기존 데이터에서 Data Augmentation 방법을 사용하면 데이터 자체에 노이즈가 생기니 이를 학습해 모델을 견고하게 할 수 있음



고양이 이미지를 뒤집거나 각도를 회전 시킨다고 개가 되는 것은 아니다!

• Introduction

- 이 논문에서 제시하는 문제
  - 자연어 처리에서는 컴퓨터 비전의 방법을 사용할 수 없음
    - 단어의 뜻만 바뀌어도 문장의 의미 자체가 변함!
      -> 섣불리 만지면 오히려 정확도가 떨어짐

  - 이전에 제안된 NLP Data Augmentation 기법 또한 최대한 의미를 손상시키지 않는 범위 내에서 이뤄짐
    ① 문장을 프랑스어로 번역하고 다시 영어로 번역해서 새로운 데이터를 얻는 방식 (Back-translation)
    ② 데이터에 노이즈를 가볍게 주는 방식 (data noising)
    ③ 유의어로 교체해주는 예측 언어 모델

  - 본 테크닉들이 유효하지만 성능 대비 구현 비용이 높아서 잘 사용되지는 않음
  - 그래서 본 논문은 EDA라 불리는 보편적인 NLP Data Augmetation을 소개

• EDA

- Method of EDA
  ① SR(Synonym Replacement) : 문장에서 랜덤으로 stop words가 아닌 n개의 단어들을 선택해 임의로 선택한 동의어들 중 하나로 바꾸는 기법
  ② RI(Random Insertion) : 문장 내에서 stop word를 제외한 나머지 단어들 중, 랜덤으로 선택한 단어의 동의어를 임의로 정함. 그리고 동의어를 문장 내 임의의 자리에 넣는걸 n번 반복
  ③ RS(Random Swap) : 무작위로 문장 내에서 두 단어를 선택하고 위치를 바꿈. 이것도 n번 반복
  ④ RD(Random Deletion) : 확률 p를 통해 문장 내에 있는 각 단어들을 랜덤하게 삭제

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A ***lamentable***, superior human comedy played out on the ***backward*** road of life.
RI	A sad, superior human comedy played out on ***funniness*** the back roads of life.
RS	A sad, superior human comedy played out on ***roads*** back ***the*** of life.
RD	A sad, superior human out on the roads of life.

Table 1: Sentences generated using EDA. SR: synonym replacement. RI: random insertion. RS: random swap. RD: random deletion.

• EDA

- 긴 문장은 짧은 문장보다 단어가 많으니, 원래의 라벨을 유지하면서 노이즈에 상대적으로 영향을 덜 받음

- 이 기준을 수식화한게 $n=\alpha l$
  - 문장의 길이에 따라 n이 변경되도록 문장의 길이(l)을 수식에 반영
  - $n=\alpha l$ (α: 문장 내 변경되는 단어의 비율, l: 문장의 길이)
  - 각 문장마다 n개의 augmented 문장을 생성하도록 함
  - RD에서는 $p = \alpha$

• Experimental Setup

- 실험에는 5개의 benchmark text classification task를 사용
  - SST-2: Stanford Sentiment Treebank (Socher et al., 2013)
  - CR: customer reviews (Hu and Liu, 2004; Liu et al., 2015)
  - SUBJ: subjectivity/objectivity dataset (Pang and Lee, 2004
  - TREC: question type dataset (Li and Roth, 2002)
  - PC: Pro-Con dataset (Ganapathibhotla and Liu, 2008)

Dataset	c	l	N_{train}	N_{test}	$\|V\|$
SST-2	2	17	7,447	1,752	15,708
CR	2	18	4,082	452	6,386
SUBJ	2	21	9,000	1,000	22,329
TREC	6	9	5,452	500	8,263
PC	2	7	39,418	4,508	11,518

Table 5: Summary statistics for five text classification datasets. c: number of classes. l: average sentence length (number of words). N_{train}: number of training samples. N_{test}: number of testing samples. $|V|$: size of vocabulary.

• Experimental Setup

- Data Augmentation이 더 작은 데이터 집합에 도움이 된다고 가정하고, 랜덤 샘플링을 통해 N_{train} = {500, 2,000, 5,000, all available data} 총 네 가지로 테스트 진행

- 사용한 Classification method
  - LSTM-RNN (Liu et al., 2016)
  - CNN (Kim, 2014)

• Results

- 5개의 task에 대해 각 method의 평균 Accuracy를 측정
- 평균적으로 전체 데이터셋을 사용할 경우 0.8%의 성능 향상이, N_{train} = 500 일 때는 3.0% 정도 상승

	Training Set Size			
Model	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**

Table 2: Average performances (%) across five text classification tasks for models with and without EDA on different training set sizes.

• Results

- 데이터 셋을 몇 퍼센트 쓰면서 EDA를 사용했을 때 정확도가 얼마나 상승했는지 그래프로 나타냄
- 가설대로 기존 데이터셋의 비율이 적으면 EDA로 늘린 데이터셋과의 정확도가 커지고, 비율이 커질수록 차이가 작아짐

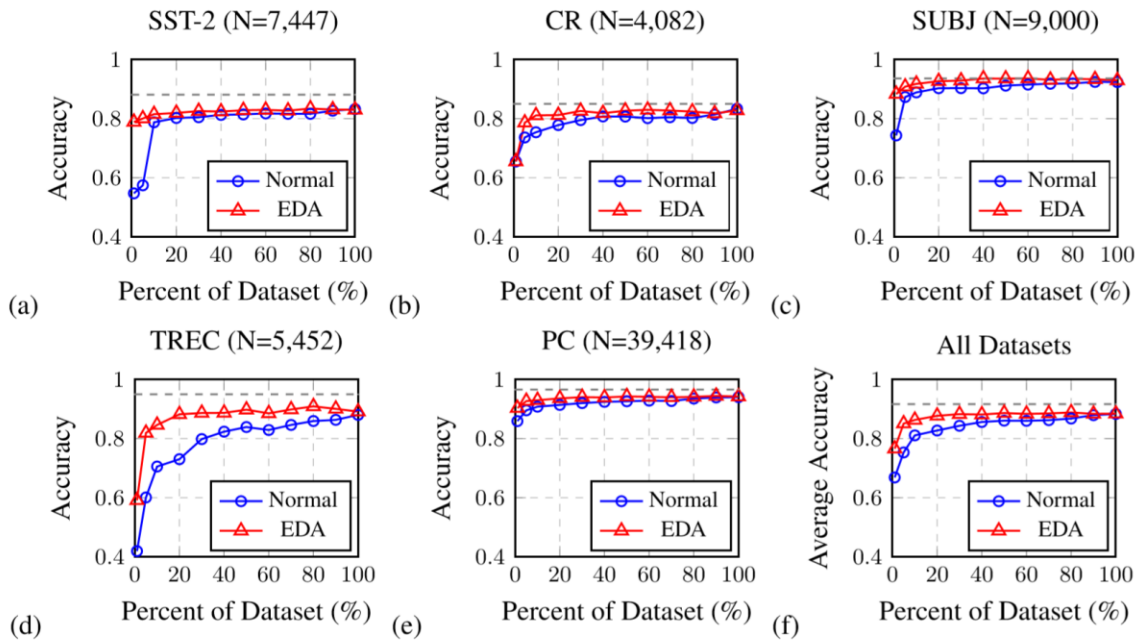
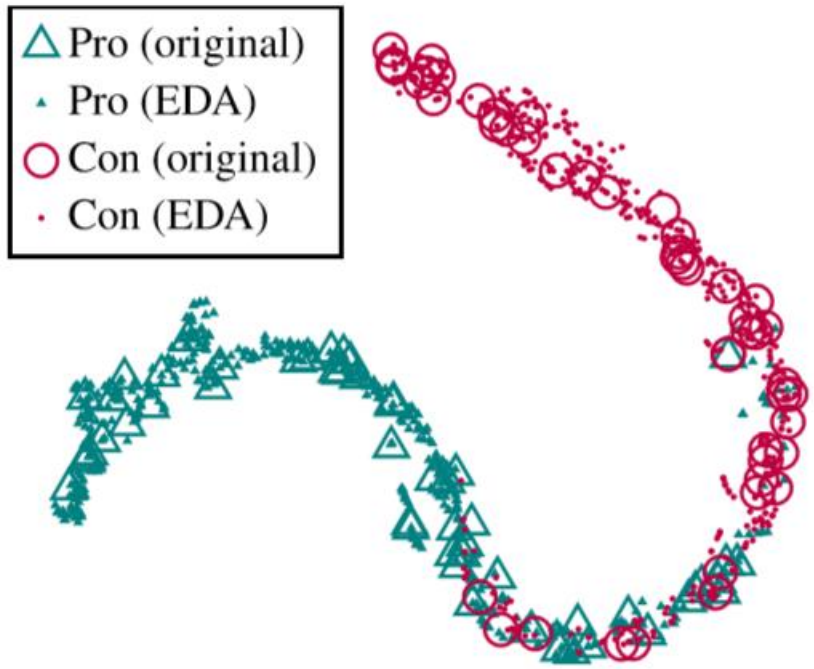


Figure 1: Performance on benchmark text classification tasks with and without EDA, for various dataset sizes used for training. For reference, the dotted grey line indicates best performances from Kim (2014) for SST-2, CR, SUBJ, and TREC, and Ganapathibhotla (2008) for PC.

• Results

- 기존 NLP Data Augmentation의 문제점
  -> 자연어 데이터를 섣불리 만지면 오히려 성능이 떨어짐

- EDA를 사용해 새로 만들어낸 문장이 문장의 라벨을 보존했는가?
  -> PC(Pro-Con dataset) dataset에 대해 EDA를 적용하고 T-SNE로 시각화
  -> 생성된 문장의 latent vector가 동일 label을 가진 원본 문장들과 가까운 위치에 있음



• Results

- 파라미터 α에 대한 성능 분석
  ① SR은 작은 α에 대해 잘 동작 -> 너무 많은 단어를 교체하면 기존 문장과 의미가 달라짐
  ② RI는 α에 대해 성능이 안정적 -> 원래 문장과 문장순서가 유지
  ③ RS는 α < 0.2 일 때 높은 성능을 보이고, 그 이후는 줄어듦 -> 문장 내 단어 순서를 지나치게 바꾸면 문법, 의미 상실
  ④ RD는 작은 α에 대해 잘 동작 -> 문장 내 단어들이 너무 삭제되면 의미 상실

- 대체로 모든 방법들에 대해 α = 0.1이 좋은 성능을 보였음(sweet spot)

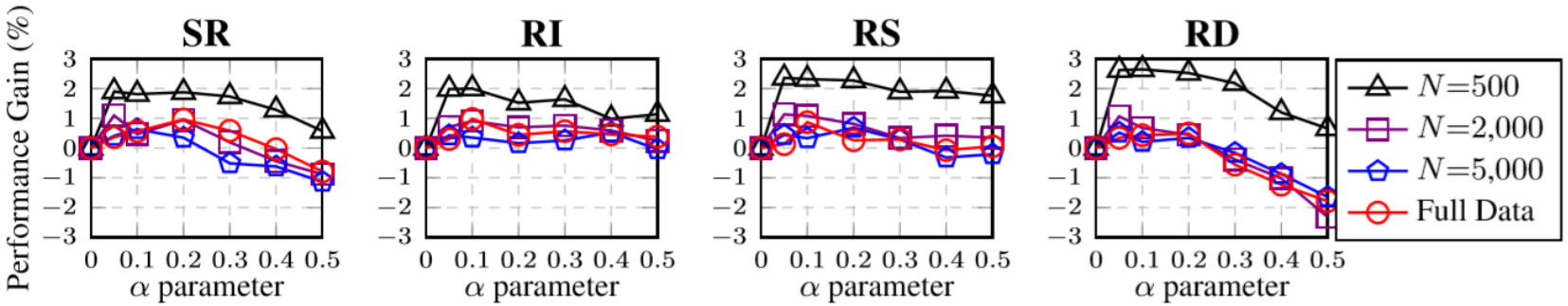


Figure 3: Average performance gain of EDA operations over five text classification tasks for different training set sizes. The α parameter roughly means "percent of words in sentence changed by each augmentation." SR: synonym replacement. RI: random insertion. RS: random swap. RD: random deletion.

• Results

- N에 대한 성능 분석
  - n={1, 2, 4, 8, 16, 32} 로 실험하여 데이터셋의 크기별 최적의 α, n 값에 대해 제시
  - 적은 데이터셋에서는 오버피팅이 발생할 가능성이 더 높기 때문에 많은 문장을 augmentation 할수록 더 큰 성능 향상을 보임
  - 반대로 큰 데이터셋에서는 비슷한 데이터를 늘려봐야 오버피팅 될 가능성이 높아지기 때문에 문장당 4개 이상의 augmentation sentence를 추가하는 건 도움이 되지 않는다

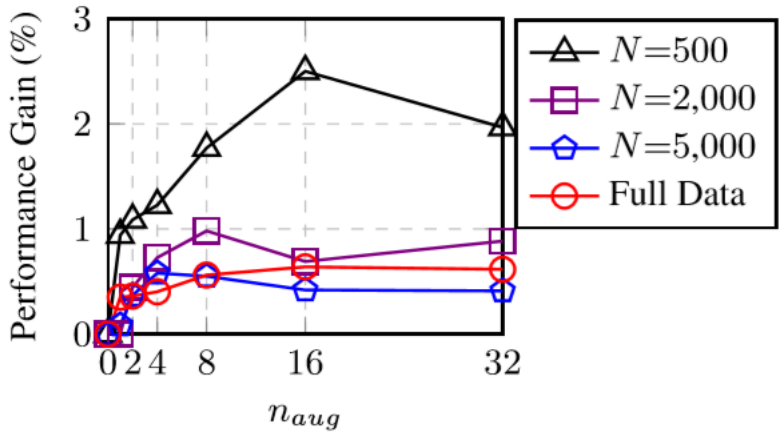


Figure 4: Average performance gain of EDA across five text classification tasks for various training set sizes. n_{aug} is the number of generated augmented sentences per original sentence.

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

Table 3: Recommended usage parameters.

• Discussion and Limitations

- 데이터가 충분할 때 성능적으로 이득이 제한적일 수 있음
  - 다섯 가지 벤치마크에서 모든 데이터셋에 대해 EDA를 적용했을 경우 1%의 정확도 상승

- 작은 데이터셋에서 더 좋은 성능을 얻는 건 명백하지만 프리트레인 모델(ELMo, BERT)을 사용하는 경우에는 개선 효과가 무시해도 되는 수준

• Conclusion

- 기존 연구들과 비교했을 때 언어모델이나 Extra Data가 필요하지 않는다는 것은 장점
- 소규모 데이터셋에 대한 학습을 진행할 때 성능을 실질적으로 향상시키고 오버피팅을 감소시킴

Technique (#datasets)	LM	Ex Dat
Trans. data aug.[1] (1)	yes	yes
Back-translation[2] (1)	yes	yes
VAE + discrim.[3] (2)	yes	yes
Noising[4] (1)	yes	no
Back-translation[5] (2)	yes	no
LM + SR[6] (2)	yes	no
Contextual aug.[7] (5)	yes	no
SR - kNN[8] (1)	no	no
EDA (5)	**no**	**no**

Table 4: Related work in data augmentation. #datasets: number of datasets used for evaluation. Gain: reported performance gain on all evaluation datasets. LM: requires training a language model or deep learning. Ex Dat: requires an external dataset.[9]