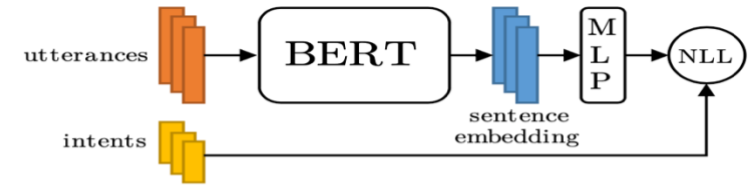
제목 : A Closer Look At Feature Space Data Augmentation For Few-Shot Intent Classification

1. Introduction
    a. 기존 챗봇에서는 데이터를 수집하고 정제하는 것이 어렵다. 특히 새로운 Intent 에 작은 Example 로는 classification 하는데 성능이 떨어지고, 이를 Few-Shot Integration 문제라 한다.
    b. 따라서 이를 해결하기 위해서 6 Feature Space Data Augmentation method 를 분석하고, Bert 와 같은 Supervised and Unsupervised Method 를 조합한 FSI 세팅을 사용
        i. 보통에 이런 연구에서 UnRealistic 하게 작은 카테고리에 작은 example 로 실험하기 때문에 현실에 맞지 않다
        ii. 여기서는 좀더 현실에 맞는 FSI 에 대해서 설명
        iii. 여기 논문에서는 제한된 데이터 상황에서 Intent Classification Performance 를 향상하기 위한 Feature Space Data Augmentation(FDA) 에 대해서 집중함
        iv. 여기서는 6 개의 FDA method 에 대해서 Study 함
            1. up- sampling in the feature space UPSAMPLE – no training
            2. random perturbation PERTURB – no training
            3. extrapolation EXTRA – no training
            4. conditional variational auto-encoder “Conditional VAE” **– Deep Learning**
            5. delta encoder that have been especially designed to work in the few-shot learning setting DELTA **– Deep Learning**
            6. linear delta which is a linear version of the delta encoder LINEAR. – no training
        v. BERT 에 Linear Data Augmentation 을 조합하는 것이 엄청 효과가 있다는 것을 증명
    c. SNIPS 와 Facebook Dialog corpus 를 이용한 실험에서 Data Augmentation Feature Space 가 효과적으로 Traditional Transfer Learning 보다 Intent 를 분류하는데 효과적이라는 것을 증명한다.
    d. 특히 1) Intent Space 안에서 Upsampling 하는 것이 feature space augmentation 을 위한 의미있는 Baseline 이 되고, 2) 새로운 예제에 두 예제 간에 차이를(차이점을 Feature 로 표현) 더하는 것이 쉽지만, 효과적인 Data augmentation 이란 것을 보여준다

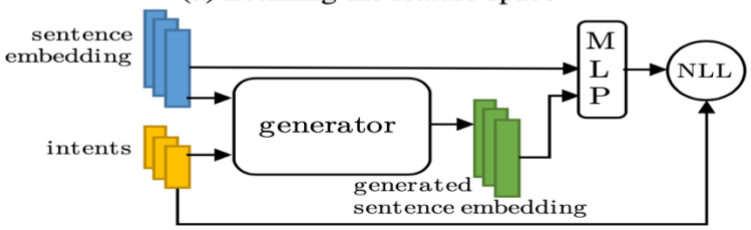
e. Contribution
    i. FSI 기술 평가 및 비교
    ii. 다양한 FDA 비교 및 평가 (항상 Deep Learning 모델이 좋은건 아니다)
    iii. FSI 에 적용할 FDA 를 추천(BERT + FSI) 가이드라인

2. Related Work
    a. Few-shot learning
        i. 이전 Computer Vision 분야에서 사용, Embedding(Feature) space 를 먼저 학습하고, 아주 작은 training example 로 새로운 카테고리의 Instance 를 simple metric 으로 분류
        ii. 추가적인 Meta-learning Based Approach 인 Metric-learning 등이 있음
        iii. 최근에는 Text Data 를 Few-shot learning 을 사용
    b. Generative Model
        i. Data Augmentation 을 이용한 Image Classification 에 사용하는데, 이는 GAN 이나, Auto-Encoder 모델을 이용해서 데이터를 더 많이 생성하는 것
        ii. 하지만 데이터가 충분히 많이 있어야 GAN 이나 Auto-Encoder 학습이 잘되고 데이터 생성이 잘된다.
        iii. 이를 극복하기 위해서 Feature Space 에서 Training data 의 Augment 하는 방법을 제안
        iv. 기존 데이터를 이용한 새로운 분류 라벨 데이터에 포함 및 재생성
        v. 하지만 Text 생성은 discreate sequence 이므로 이를 학습하는데 문제가 있음 또한 Training Data 가 많아야 함(GPT 모델도 있음)
        vi. 따라서 이 논문에서는 Generative model 생성에 초점을 맞추고 Feature Space 에 대한 Data Augment 에 대해서 알아봄(Text Classification 위한 FSI 문제를 해결하기 위해서)

3. Data Augmentation in Feature Space
    a. Introduction
        i. 1) Data Representation + Feature Extractor 학습
        ii. 2) Feature Space 에서 lower Resource class 에 대한 new data 생성
        iii. 3) Data 를 생성한 이후, Classifier 는 Real 과 Augmented data 를 학습



(a) Learning the feature space



(b) Learning with augmented data

iv.

v. BERT 에서 Text Representation 뽑음(BERT 를 안다고 가정)

b. Data Augmentation 을 위해서 Six Different FDA Method 를 적용하여 New Example 을 Feature Space 로 생성. 그리고 1-layer Softmax classifier Intent 를 분류

c. 6 FDA Method

i. Upsampling(ex 이미지 5x5 -> 7x7 convolution upsampling)

1. 기존 데이터를 Upsampling 하여 데이터를 늘리는 방법, FDA Techniques 에 효과적인 좋은 Baseline 이 됨

ii. Random Perturbation(랜덤한 혼돈?)

1. 기존 데이터에 Noise 추가 하여 데이터 생성
2. 여기 실험에서는 additive and Multiplicative Perturbation 를 사용

iii. Conditional VAE

1. Latent Distribution 으로 부터 샘플링을 통한 New Data 생성
2. 기본적인 Encoder – Decoder 모델이면 여기서 Encoder 가 Latent Vector Z Feature 를 추출하고, 이를 바탕으로 Decoder 에서 데이터를 생성

iv. Linear Delta

1. Example Pair 의 차이점을 학습하고, 이를 다른 Example 에 추가하는 것

$$\hat{X} = (X_i - X_j) + X_k$$

a.

v. Extrapolation(외삽법: 수학에서 원래의 관찰 범위를 넘어서서 다른 변수와의 관계에 기초하여 변수의 값을 추정하는 과정)

1. $$\hat{X} = (X_i - X_j) * \lambda + X_i$$

vi. Delta-Encoder

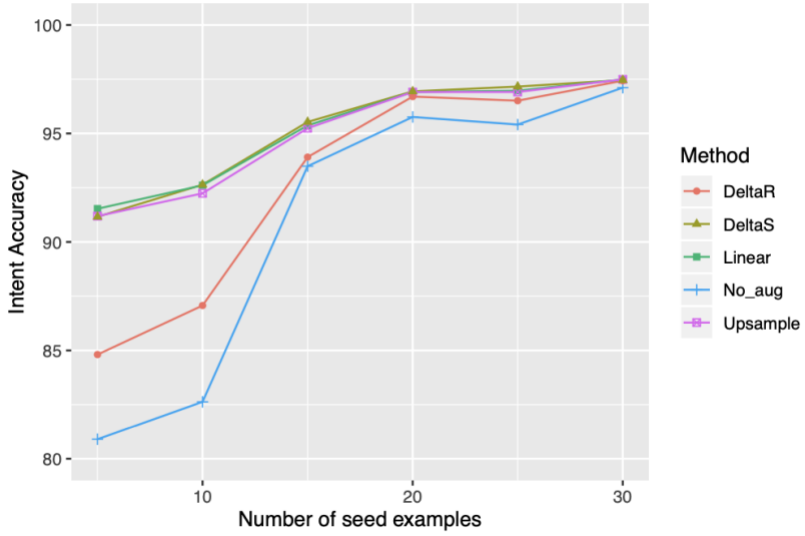
1. Deformations(Deltas) : 2 개의 Class 에서 example 의 차이점
2. Autoencoder 베이스 모델로 Transferable Deformation 을 2 개의 다른 Source Class 로 부터 학습하고, 이를 새로운 클래스의 Few Example 합성 -> 새로운 데이터 생성
3. Data 생성을 위한 2 개의 다른 source sentence pair 를 선택하는 방법
    a. 1)DeltaR:
        i. Random 하게 class 를 선택해서 Sentence Pair 구축 , New example 을 합성하기 위해서 Multiple source categories 로 부터 델타 적용
    b. 2)DeltaS:
        i. Target category 로 부터 Sentence Pair 선택, Same target category 로 부터 델타 적용

4. Experiment
    a. Dataset
        i. Public data
            1. SNIPS, Facebook Dialog(FBDialog)Corpus -> Train, Dev, Test
                a. SNIPS : 7 intent(균형)
                b. FBDialog: 데이터 셋 불균형 (Intent 당 Maximum : 8,860~ Minimum 4)
        ii. Simulating Few-shot integration
            1. New Intent 는 Random Sampling
            2. Dev set 에 Target intent Data 제거
            3. Different Augmentation Method 로 100, 512 Example 생성
            4. 결과의 임의 변동을 설명하기 위해서 Target intent 를 10 번 테스트하고 평균과 표준 편차를 Report 함
5. Result and Discussion

Size	Method	SNIPS	FBDialog
No Augmentation		98.14 (0.42)	94.99 (0.18)
5%	UPSAMPLE	98.14 (0.47)	95.01 (0.16)
	PERTURB	**98.26** (0.40)	94.98 (0.19)
	LINEAR	98.14 (0.45)	**95.02** (0.21)
	EXTRA	98.14 (0.45)	**95.02** (0.20)
	CVAE	98.14 (0.45)	94.98 (0.24)
	DELTAR	98.23 (0.46)	95.00 (0.22)
	DELTAS	**98.26** (0.42)	95.00 (0.20)
10%	UPSAMPLE	98.14 (0.47)	94.94 (0.18)
	PERTURB	98.23 (0.41)	94.98 (0.24)
	LINEAR	98.09 (0.50)	**95.02** (0.18)
	EXTRA	98.11 (0.49)	95.01 (0.19)
	CVAE	98.20 (0.42)	94.99 (0.26)
	DELTAR	**98.26** (0.42)	94.99 (0.21)
	DELTAS	98.23 (0.42)	94.97 (0.22)
20%	UPSAMPLE	98.14 (0.45)	95.02 (0.12)
	PERTURB	98.14 (0.44)	94.99 (0.20)
	LINEAR	98.17 (0.43)	95.05 (0.23)
	EXTRA	98.14 (0.45)	95.07 (0.11)
	CVAE	98.11 (0.44)	94.98 (0.23)
	DELTAR	**98.26** (0.40)	**95.08** (0.19)
	DELTAS	98.20 (0.46)	95.04 (0.22)

Table 1: IC accuracy on SNIPS and Facebook dataset with all training data, reported as *mean (SD)*.

a.
b. 5.1 FDA for Data-rich Classification
    i. 데이터를 생성하는 것을 5, 10, 20 %증가 시키면서 실험
c. 5.2 Impact Of the Number of Seed Examples



    i.
d. 5.3 Few-shot Integration

e. 5.4 Upsampling: Text Space vs Latent Space

#	Method	SNIPS	FBDialog
No Augmentation		87.46(2.87)	81.29(0.11)
100	UPSAMPLE	94.26(1.66)	**84.34**(1.84)
	PERTURB	94.18(1.74)	84.04(1.95)
	CVAE	94.10(1.83)	84.10(1.94)
	LINEAR	**94.36**(1.69)	84.31(1.9)
	EXTRA	94.30(1.68)	84.13(1.83)
	DELTAR	91.32(3.12)	81.97(0.76)
	DELTAS	94.28(1.92)	83.50(1.92)
512	UPSAMPLE	95.68(0.86)	89.03(0.99)
	PERTURB	95.65(0.92)	89.02(0.99)
	CVAE	95.46(1.03)	88.71(1.09)
	LINEAR	**95.87**(0.87)	**89.30**(1.03)
	EXTRA	95.82(0.89)	89.21(0.99)
	DELTAR	95.33(1.56)	87.28(1.46)
	DELTAS	**95.88**(1.04)	89.15(1.12)

Table 2: Average IC accuracy for all intents' FSI simulations on SNIPS and FBDialog dataset. For each simulation, $k = 10$ seed examples are used for target intent. Scores are reported as *mean (SD)*. Refer to Appendix's Table 5 and Table 6 for individual intents' results.

#	Method	Overall Mean
No Augmentation		94.38(1.23)
100	UPSAMPLE	94.53(1.12)
	PERTURB	94.52(1.18)
	CVAE	94.53(1.18)
	LINEAR	94.53(1.12)
	EXTRA	94.53(1.13)
	DELTAR	**94.62**(1.16)
	DELTAS	94.57(1.14)
512	UPSAMPLE	94.67(1.11)
	PERTURB	94.68(1.14)
	CVAE	94.73(1.11)
	LINEAR	94.67(1.11)
	EXTRA	94.67(1.11)
	DELTAR	**94.88**(1.12)
	DELTAS	94.74(1.12)

Table 3: IC accuracy on SNIPS dataset in the FSI setting, reported as *mean (SD)*. The 10 seed examples are upsampled to 100 to train the feature extractor. Refer to Appendix's Table 7 for individual intents' results.

i.

ii. Upsampling 이 Text Space 적용시 augmentation baseline 87.46 -> 94.38 향상

iii. Upsampling 이 Text Sapce 보다 Latent Space 에서 더 효과적임

1. BERT 모델이 들어간 DELTA R, S 에서 Text Space 에 Upsmapling 한 결과가 낮은 것을 볼 수 있고 95.88 -> 94.88
2. Text space 에서 Upsampling 이 Overfitting 되었다고 가설을 세움

f. Effect Of the Pre-trained BERT Encoder(Fine-tuning book intent)

#	Method	Playlist	Restaurant	Weather	Music	Book	Work	Event	Overall Mean
No Augmentation		82.63(5.11)	87.86(3.53)	84.51(1.3)	88.07(2.37)	96.81(2.94)	85.14(1.53)	87.19(3.31)	87.46(2.87)
100	UPSAMPLE	92.24(2.96)	97.7(0.67)	96.44(0.75)	94.57(1.1)	97.96(0.82)	89.61(3.01)	91.26(2.35)	94.26(1.66)
	PERTURB	**93.09**(2.55)	97.41(0.92)	96.07(1.35)	94.39(1.13)	97.86(0.93)	89.36(2.76)	91.09(2.53)	94.18(1.74)
	CVAE	92.4(3.66)	97.47(0.67)	96.49(1.07)	94.36(1.26)	97.71(1.1)	89.1(2.79)	91.2(2.22)	94.1(1.83)
	LINEAR	92.61(3.02)	97.74(0.67)	96.44(0.77)	94.63(1.18)	**97.97**(0.78)	**89.61**(3.05)	91.53(2.34)	**94.36**(1.69)
	EXTRA	92.36(3.0)	97.74(0.66)	96.41(0.77)	94.6(1.18)	97.97(0.78)	89.47(3.11)	91.51(2.3)	94.3(1.68)
	DELTAR	87.07(4.67)	93.57(4.07)	91.0(4.23)	94.87(1.28)	97.66(1.42)	85.97(2.34)	89.11(3.84)	91.32(3.12)
	DELTAS	92.64(4.49)	**97.76**(0.7)	96.41(1.25)	**94.99**(0.92)	97.83(0.99)	88.69(2.69)	**91.64**(2.36)	94.28(1.92)
512	UPSAMPLE	95.3(1.09)	98.0(0.64)	97.63(0.34)	95.57(0.87)	98.03(0.55)	92.0(1.49)	93.26(1.05)	95.68(0.86)
	PERTURB	95.33(1.2)	97.94(0.6)	97.6(0.44)	95.5(0.91)	97.91(0.55)	92.03(1.78)	93.21(0.99)	95.65(0.92)
	CVAE	95.46(1.12)	97.89(0.62)	97.54(0.43)	95.36(1.02)	97.93(0.7)	91.34(2.17)	92.73(1.19)	95.46(1.03)
	LINEAR	95.39(1.1)	**98.0**(0.64)	97.67(0.36)	95.74(0.89)	**98.04**(0.5)	**92.61**(1.47)	93.66(1.13)	**95.87**(0.87)
	EXTRA	95.36(1.17)	98.0(0.64)	97.66(0.37)	95.74(0.88)	98.04(0.5)	92.29(1.52)	93.63(1.17)	95.82(0.89)
	DELTAR	95.36(1.74)	97.81(0.69)	97.6(0.44)	95.9(0.97)	97.74(1.02)	90.27(3.44)	92.61(2.64)	95.33(1.56)
	DELTAS	**95.66**(1.18)	97.96(0.59)	**97.8**(0.45)	**95.91**(0.88)	97.91(0.74)	92.26(2.57)	**93.66**(0.86)	**95.88**(1.04)

Table 5: IC accuracy on SNIPS dataset in the FSI setting ($k = 10$), reported as *mean (SD)*.

i.

Size	Method	SNIPS's AddToPlaylist		FBDialog's GetDirections	
seed examples (k)		10	100*	10	100*
No Augmentation		80.07 (2.08)	90.17 (1.39)	87.44 (0.12)	87.94 (0.32)
100	UPSAMPLE	**88.27** (1.74)	90.61 (1.52)	88.01 (0.26)	88.17 (0.32)
	PERTURB	88.03 (1.52)	90.86 (1.39)	88.01 (0.32)	88.25 (0.31)
	LINEAR	88.14 (1.62)	91.06 (1.58)	88.05 (0.25)	88.26 (0.32)
	EXTRA	88.09 (1.57)	90.74 (1.57)	**88.10** (0.29)	88.20 (0.3)
	CVAE	**88.27** (2.08)	90.90 (1.69)	88.04 (0.24)	88.17 (0.32)
	DELTAR	82.23 (2.21)	**91.46** (1.19)	87.60 (0.23)	**88.75** (0.43)
	DELTAS	84.4 (2.74)	91.07 (1.44)	88.02 (0.22)	88.57 (0.36)
512	UPSAMPLE	91.41 (1.03)	91.61 (1.4)	88.68 (0.49)	88.40 (0.35)
	PERTURB	**91.46** (0.99)	91.73 (1.32)	88.89 (0.57)	88.56 (0.39)
	LINEAR	91.20 (1.28)	91.41 (1.52)	88.97 (0.65)	88.47 (0.33)
	EXTRA	91.26 (1.22)	91.57 (1.55)	88.85 (0.61)	88.48 (0.37)
	CVAE	91.39 (0.94)	91.44 (1.2)	89.02 (0.52)	88.48 (0.4)
	DELTAR	87.09 (2.75)	**92.97** (1.2))	88.61 (0.35)	**89.70** (0.53)
	DELTAS	89.34 (1.48)	92.00 (1.25)	**89.34** (0.4)	89.09 (0.51)

Table 4: IC accuracy on SNIPS's AddToPlaylist and FBDialog's GetDirections in the FSI setting, reported as *mean (SD)*. A 1-layer Bi-LSTM model is used as a feature extractor. 100* represents 10 seed examples are upsampled to 100 to train the feature extractor.

ii.

iii. FSI 는 좋은 sentence representations 하는 Feature Extracotr 모델이 필요한데 Bi-LSTM 으로는 한계가 있고, Upsameple 과 Perturb 보다 성능이 더 않좋게 나옴 10 개 일 때, 데이터가 100 가 되니 성능이 좀 좋아지는 것을 볼 수 있음

g. 5.6 Is Delta-Encoder Effective On Text ?

i. Image Classification 에서는 DeltaR 이 Delta S 보다 못했다(충분이 다른 클레스의 데이타 사이에 variations 를 배우지 못했음

ii. Text 의 경우 Bert 는 Intra-class variation 를 잘 학습하고, Feature Representation 하는 것을 결과를 봤을 때 알 수 있음(Delta R 의 성능이 더 좋음)

h. Qualitative Evaluation

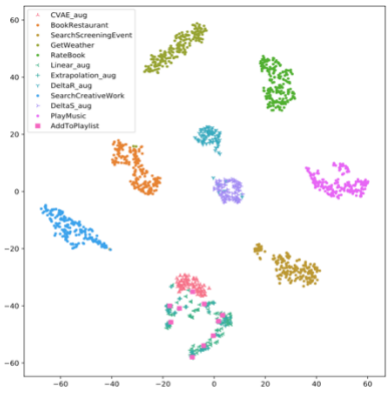


Figure 3: 10 seed examples

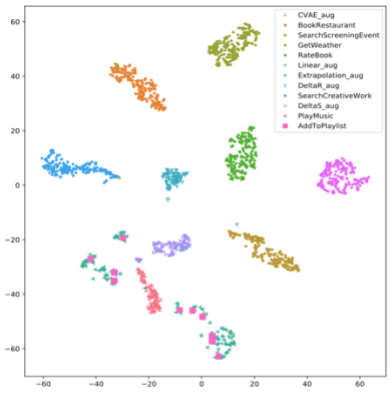


Figure 4: 10 seed examples are upsampled to 100

Figure 5: t-SNE visualization of different data augmentation methods for AddToPlaylist intent. BERT encoder is used to learn sentence representations.

i.

i.

6. Conclusion and Future Work
    a. 결국 BERT 사용한 DeltaR 기법이 FSI 에 좋은 결과를 보여주고, Text 에 Delta Encoder 는 Transferable intra-class variation 을 학습을 잘 한다는 것을 실험으로 증명
    b. 다른 NLP Test 에 사용가능

- 특이 단어
    - Extrapolation(외삽법):
        - 수학에서 원래의 관찰 범위를 넘어서서 다른 변수와의 관계에 기초하여 변수의 값을 추정하는 과정이다.
    - FSI: Few-Shot Integration
    - FDA: Feature space Data Augmentation