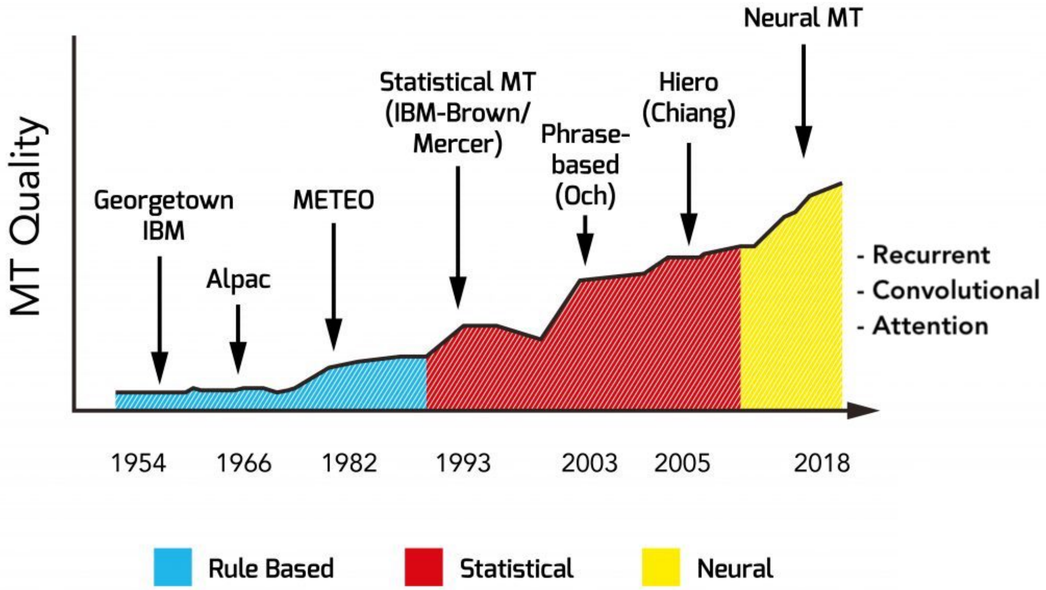
DEVIEW 2021

데이터, 정말 묻고 더블로 가? : 기계번역 성능 개선 예측 모델

박천복 NAVER Corporation/ Papago

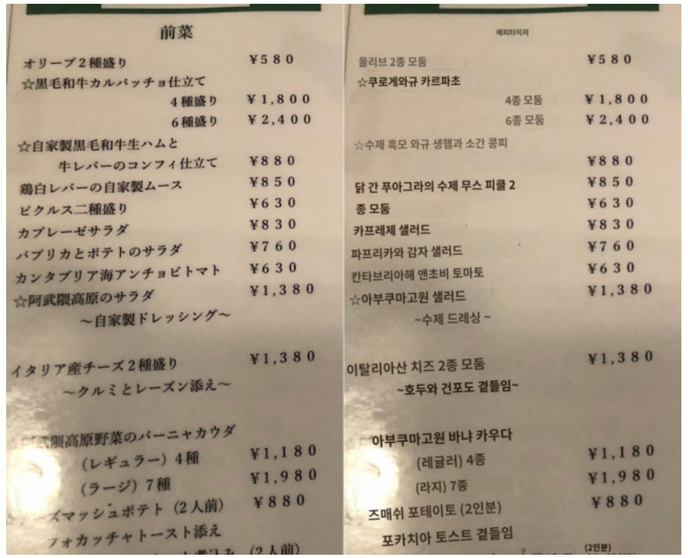
1.Motivation

1.1 일상 생활에서의 번역기



Img src: https://iconictranslation.com/what-we-do/neural-machine-translation/

여행지 숙박 리뷰 번역



여행지 메뉴판 번역 (by 파파고 이미지 바로 번역)

기계 번역 성능 개선에 따른 우리의 다양한 삶에 쓰이는 번역기

만약 이러한 일반 기계번역기에
연예/기술과 같은 전문 도메인의 문장을 넣는다면?

1.2 하나의 번역기의 한계

입력 : 이번 주말에 **막콘** 갈거에요.

I'm going to Makkon this weekend.

1.2 하나의 번역기의 한계

입력 : 이번 주말에 **막콘** 갈거에요.

I'm going to **last concert** this weekend.

1.3 전용 번역기의 필요성

우리 삶에서 번역기 사용의 확장

자신의 도메인을 위한 고성능의
전용 번역기의 필요성

일반 일상생활에서 자유롭게
사용될 하나의 번역기

각 특정/전문 도메인을 위한
여러 전용 번역기

1.4 전용 번역기 만들기

전용 번역기를 만들기 위해서 필요한 것은(Domain adaptation)?

잘 학습된 번역기

(다양한 일반적인 도메인으로 학습이 된)

전용 번역기 학습을 위한 병렬 문장들

(In-domain data)

* 잘 학습된 번역기 (앞으로, Baseline NMT라 표기)

1.5 In-domain data 생성

전용 번역기를 만들기 위해서 필요한 것은(Domain adaptation)?

잘 학습된 번역기

(다양한 일반적인 도메인으로 학습이 된)

전용 번역기 학습을 위한 병렬 문장들

(In-domain data)

우리가 새로 만들어야 하는 것

* 잘 학습된 번역기 (앞으로, Baseline NMT라 표기)

1.5 성능 그래프를 추론해야 하는 이유

DEVIEW 2021

Q) 전문가가 번역해서 단일 언어 데이터를 병렬 데이터로 만든다면?

전문가 번역은 데이터의 양에 비례하여 긴 시간과 높은 비용이 필요 (+ 예산 제한)

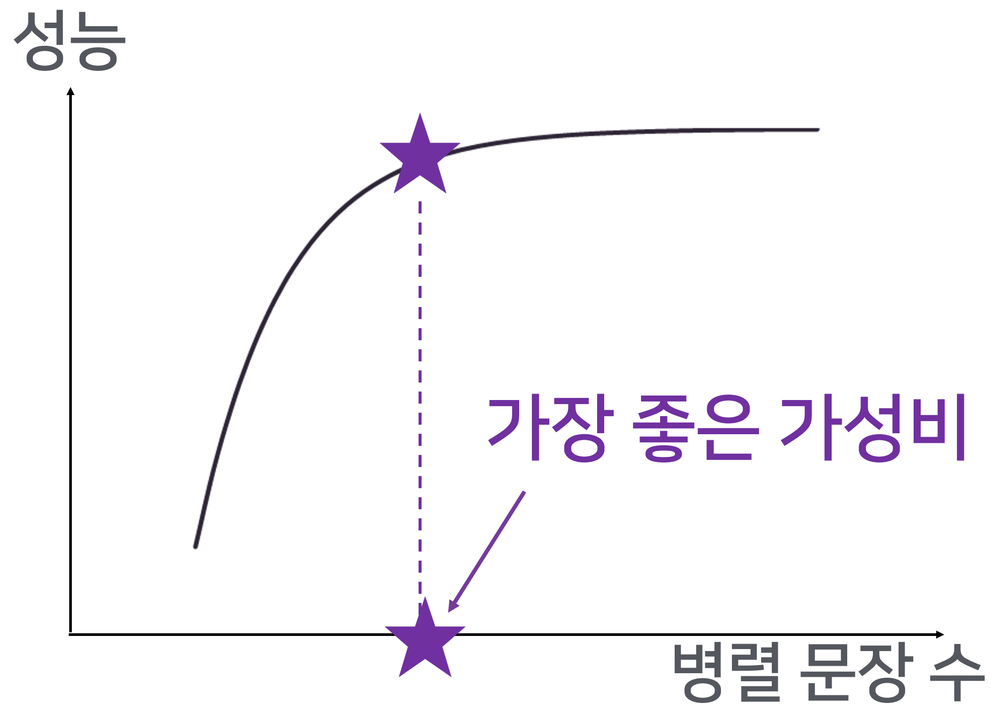
1.5 성능 그래프를 추론해야 하는 이유

Q) 전문가가 번역해서 단일 언어 데이터를 병렬 데이터로 만든다면?

전문가 번역은 데이터의 양에 비례하여 긴 시간과 높은 비용이 필요 (+ 예산 제한)

Q) 가성비가 가장 좋은 병렬 데이터란?

성능이 증가폭이 미미하기 직전의 데이터 수
또는, 원하고자 하는 성능에 해당되는 데이터 수



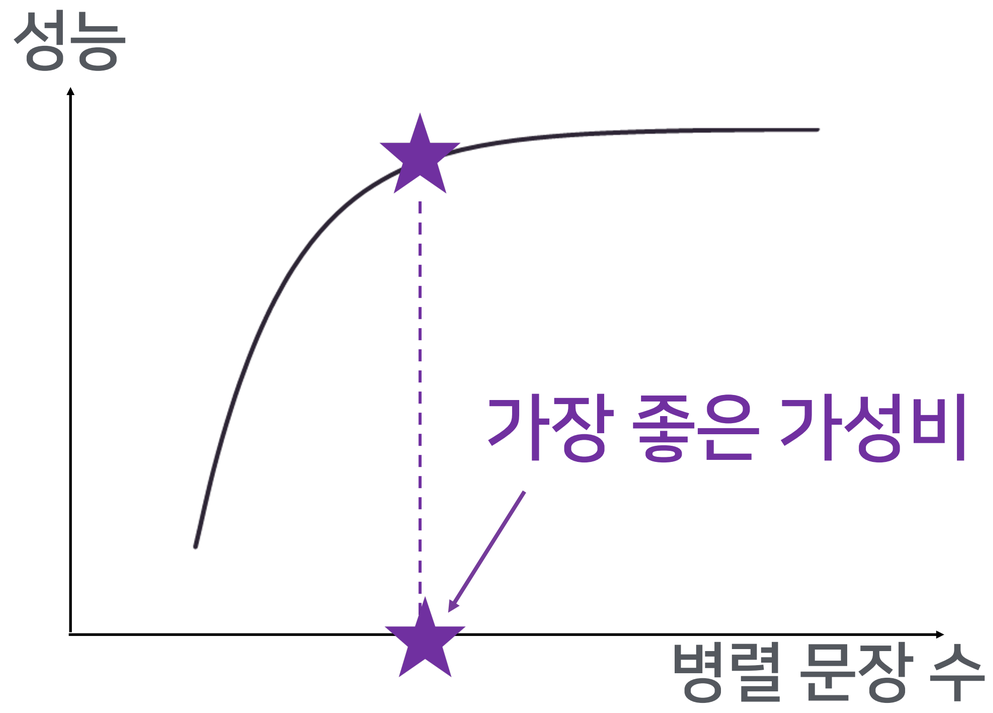
1.5 성능 그래프를 추론해야 하는 이유

Q) 전문가가 번역해서 단일 언어 데이터를 병렬 데이터로 만든다면?

전문가 번역은 데이터의 양에 비례하여 긴 시간과 높은 비용이 필요 (+ 예산 제한)

Q) 가성비가 가장 좋은 병렬 데이터란?

성능이 증가폭이 미미하기 직전의 데이터 수
또는, 원하고자 하는 성능에 해당되는 데이터 수



오른쪽의 성능 그래프는 병렬 데이터가 없으면 알 수 없다

그 전에, 성능 그래프를 추론해야 한다!

1.6 성능 그래프를 알 수 있다면

Q) “현재 모델보다 약 10%정도 성능 향상된 도메인 전용 번역 모델을 얻고 싶다면 얼마나 데이터를 만드는 비용이 들까?”

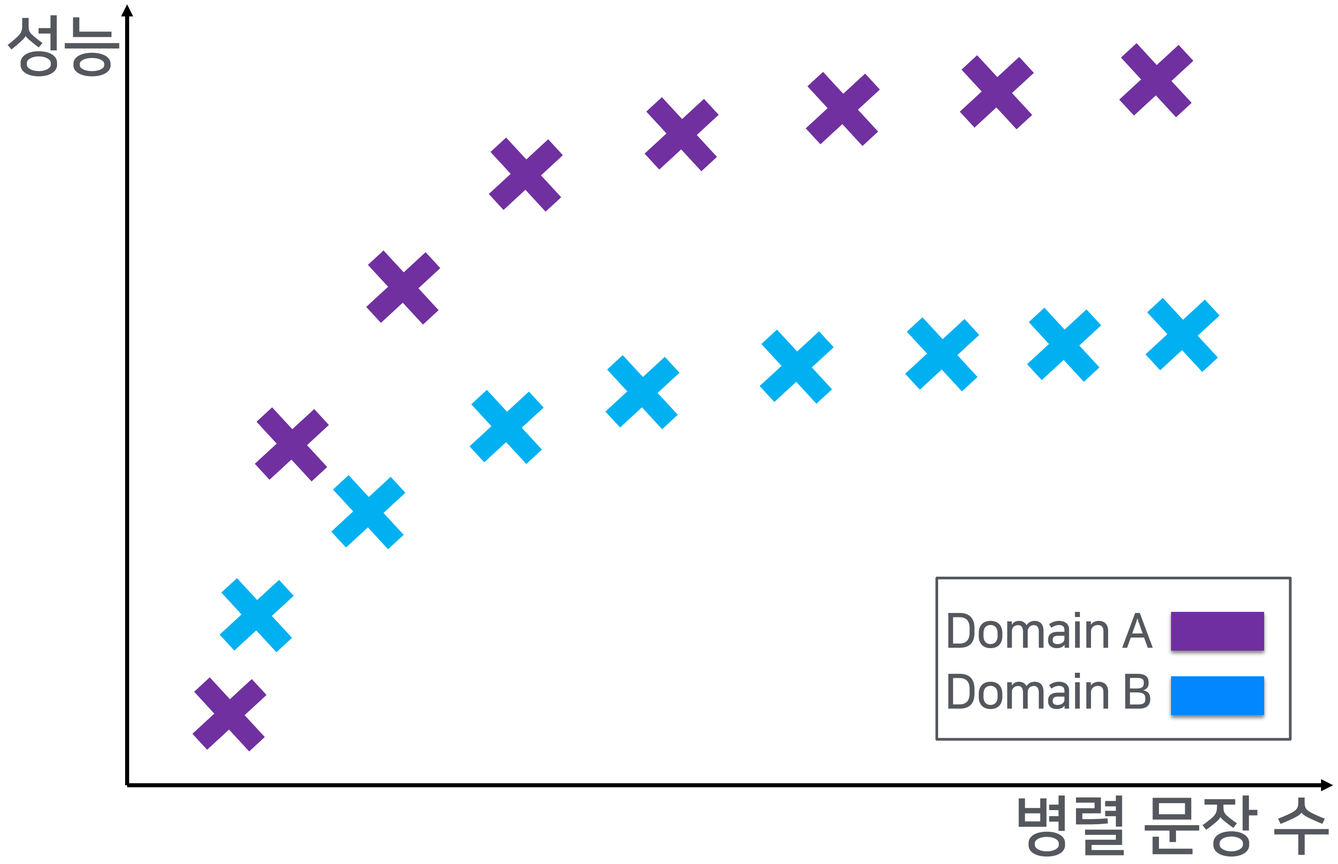
Q) “ N개의 의료 도메인 데이터를 제공해주면 의료 도메인 성능에 대해서 얼마나 향상된 모델을 얻을 수 있을까요?”

1.7 기존 방법론의 한계 및 어려움

DEVIEW 2021

Heuristic solution

다른 도메인에서 성능 개선 변화 그래프를 참고하여 내가 관심을 가지는 도메인에 대해 추론



Domain C의 성능 그래프는
어떻게 그려질까?

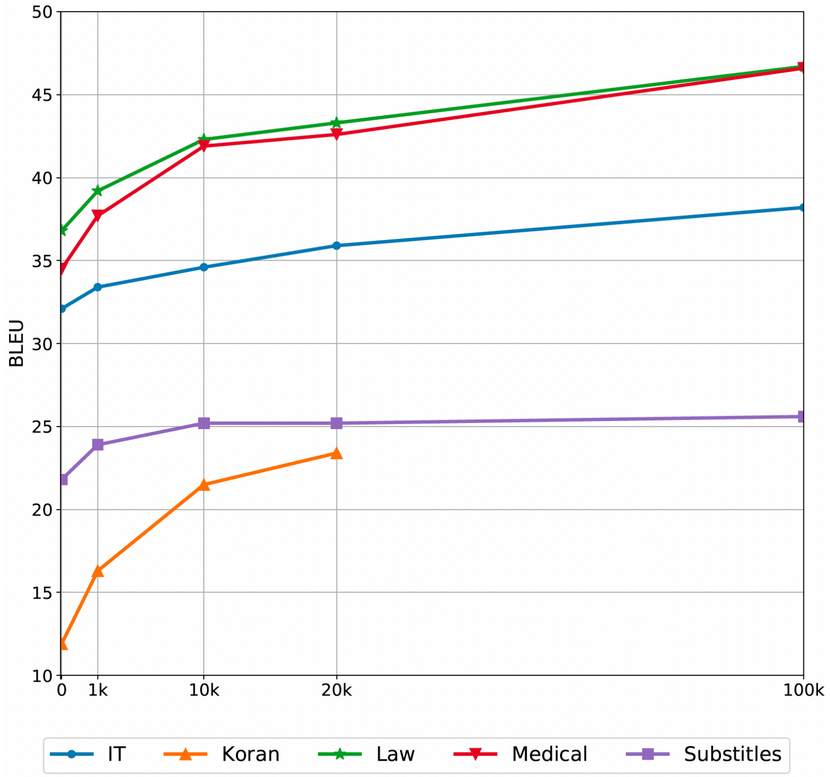
* 각 ✖ 는 finetuning후 관측된 corpus-level 성능이며 편의상 anchor point라 부름

1.7 기존 방법론의 한계 및 어려움

DEVIEW 2021

Heuristic solution

다른 도메인에서 성능 개선 변화 그래프를 참고하여 내가 관심을 가지는 도메인에 대해 추론

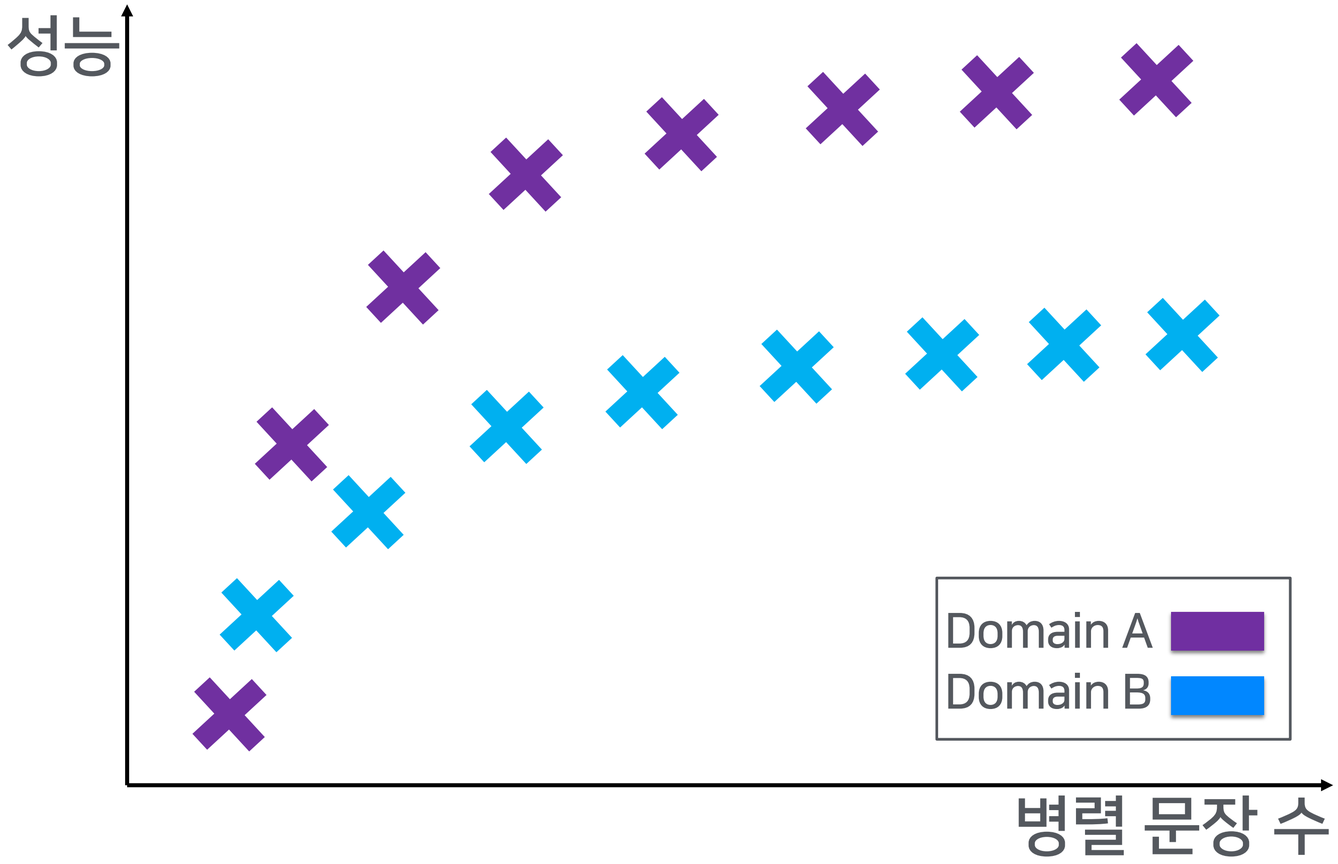


← 도메인 마다 성능 증가 폭, 초기 성능이 다름

1.7 기존 방법론의 한계 및 어려움

성능을 예측하는 Machine Learning model

각 anchor point case에 대해서 성능을 예측하는 회귀 모델을 만들기



x

(1) Domain indicator

(2) Number of sentences

Model

y

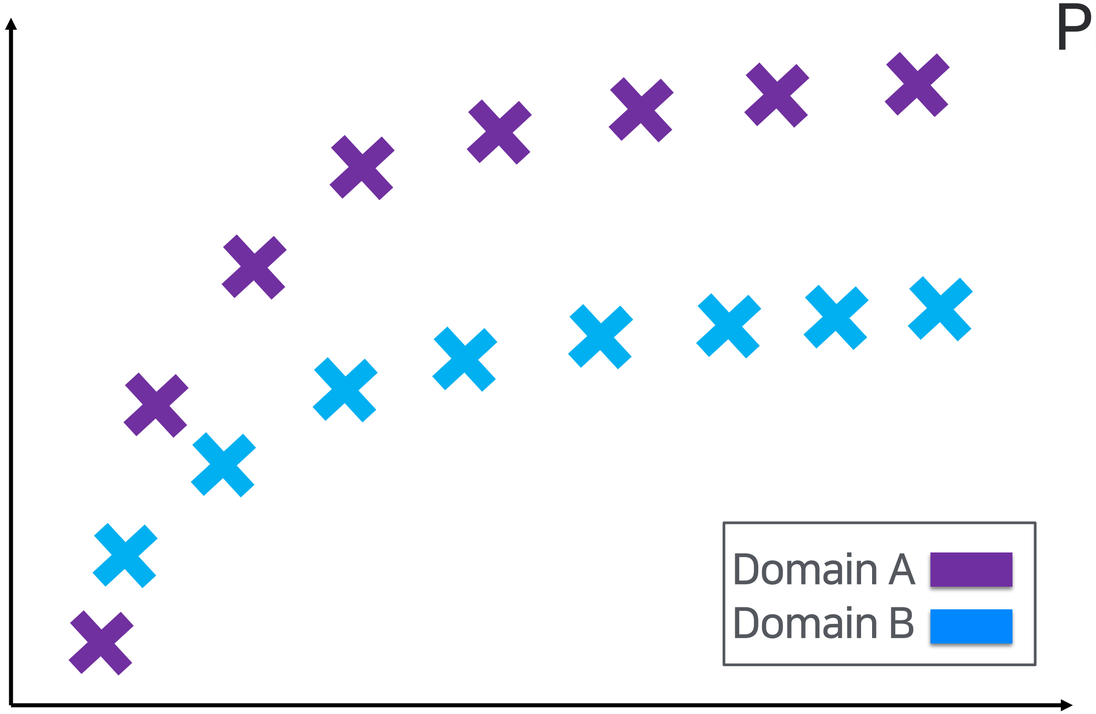
BLEU : 31.3

1.7 기존 방법론의 한계 및 어려움

DEVIEW 2021

성능을 예측하는 Machine Learning model

각 anchor point case에 대해서 성능을 예측하는 회귀 모델을 만들기



Problems in Corpus-level 회귀 모델

1) One Anchor point == one training data

- 하나의 training data를 얻기 위해서는 한번의 finetuning이 필요
- 한 번의 finetuning에 많은 computation cost와 시간이 소요됨.

2) Unseen domain에 대해서 generalization 능력이 떨어짐

1.8 Challenges

다음의 세가지 조건을 만족시켜야 함

1) 적은 training anchor points를 사용

2) 병렬 문장 없이 단일 언어 문장을 활용하여 모델이 예측

3) 임의의 도메인에서의 정확한 성능

2.Proposed Method

2.1 적은 training anchor points

다음의 세가지 조건을 만족시켜야 함

1) 적은 training anchor points를 사용

2) 병렬 문장 없이 단일 언어 문장을 활용하여 모델이 예측

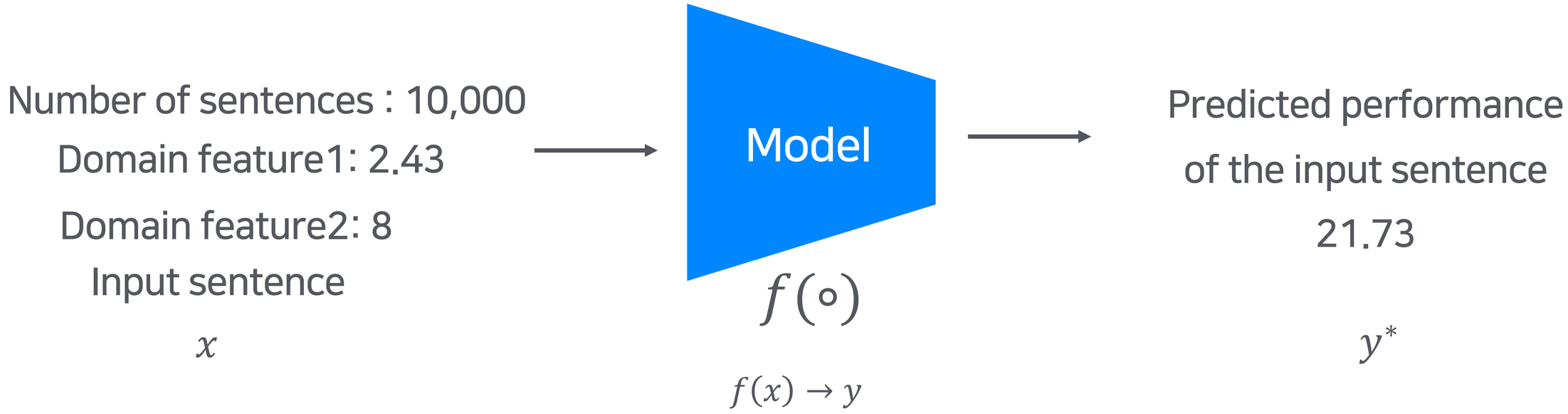
3) 임의의 도메인에서의 정확한 성능

2.1 Instance-level framework

Solution 1 : Instance-level framework

하나의 anchor point에서 N개의 training examples을 만듬

각 anchor point 별 성능을 각 문장에 대해서 예측된 성능의 총 평균 값으로 추정하게 함



y : performance

x : anchor point (* anchor point == adaptation cases)

2.1 Instance-level framework

Solution 1 : Instance-level framework

무엇이 다른가 Corpus vs Instance-level?

Corpus-level	Instance-level
1. 오직 K개의 training 데이터 얻음 2. 바로 anchor point의 성능을 예측	1. N*K개의 Training 데이터 얻음 2. 각 문장 별로 성능을 예측해서, 예측한 문장들의 성능을 평균 값으로 anchor point 성능으로 예측

* 가정 : K개의 anchor point를 training data 를 수집하기 위해 얻었다면

2.1 Instance-level framework

Solution 1 : Instance-level framework

왜 유용한가?

1. N*K개의 Training anchor point 얻음

   →상대적으로 적은 finetuning으로 모델을 학습 가능

2. 각 문장 별로 성능을 예측해서, 예측한 문장들의 성능을 평균 값으로 anchor point 성능으로 예측

   → 각 문장 별로 성능 개선 폭을 다르게 모델링 하여 문장 별 특성에 맞게 예측

2.2 Features

다음의 세가지 조건을 만족시켜야 함

1) 적은 training data anchor points를 사용

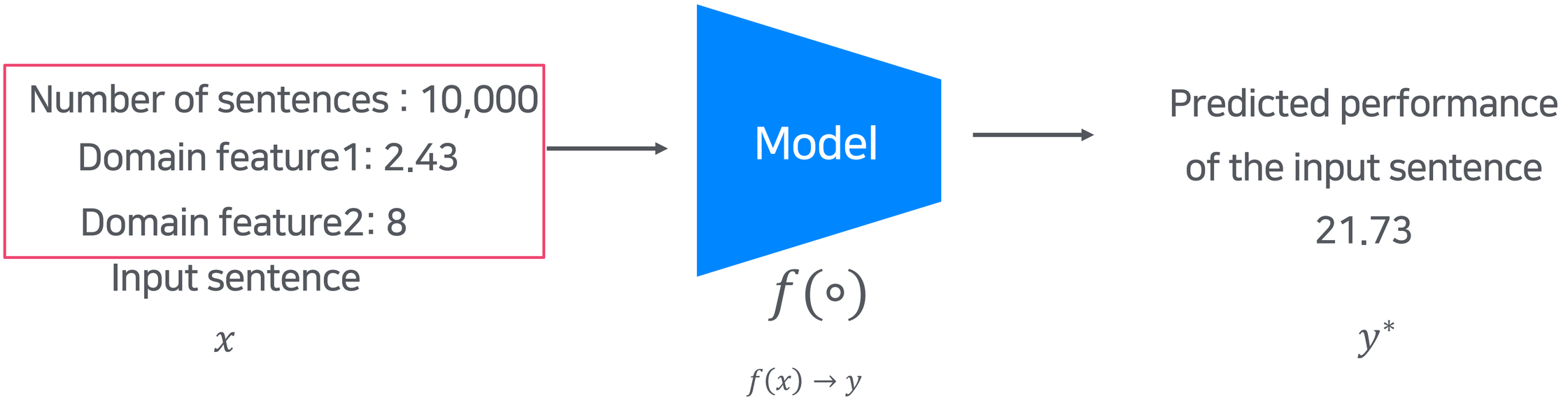
2) 병렬 문장 없이 단일 언어 문장을 활용하여 모델이 예측

3) 임의의 도메인에서의 정확한 성능

2.2 Features

Solution 2) 도메인의 특성을 정의 할 수 있는 Multiple factors를 정의

Solution 3) 단일 언어 문장만 사용하여 구할 수 있는 features들로 정의



y : performance

x : anchor point (* anchor point == adaptation cases)

2.2 Features

Domain adaptation시에 영향을 줄 것으로 예상 되는 요인:

- 문장이 가지고 있는 도메인 특성
  - 현재의 문장이 도메인 특성을 얼마나 가지는 지 general dataset 대비?
- 새로운 도메인의 난이도
  - 새로운 도메인이 baseline NMT 모델에게 얼마나 어려운지 ?
- Finetuning 하는 in-domain 데이터 안의 정보량
  - 각 anchor point의 training dataset 에는 얼마나 새로운 정보가 들어 있는지?
- Domain adaptation 알고리즘과 하이퍼파라미터 선택

2.2 Features – NMT encoder

문장이 가지고 있는 도메인 특성

방법 1) Baseline(general) NMT encoder representation을 활용

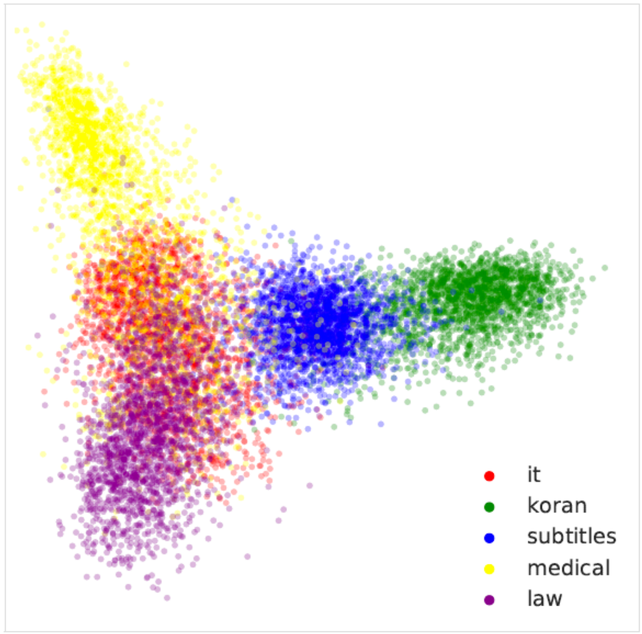


Figure 1: A 2D visualization of average-pooled BERT hidden-state sentence representations using PCA. The colors represent the domain for each sentence.

	Medical	Law	Koran	IT	Subtitles
Medical	**56.5**	18.3	1.9	11.4	4.3
Law	21.7	**59**	2.7	13.1	5.4
Koran	0.1	0.2	15.9	0.2	0.5
IT	14.9	9.6	2.8	**43**	8.6
Subtitles	7.9	5.5	6.4	8.5	27.3
All	53.3	57.2	**20.9**	42.1	**27.6**

Table 4: SacreBLEU (Post, 2018) scores of our baseline systems on the test sets of the new data split. Each row represents the results from one model on each test set. The best result in each column is marked in bold.

- Goldberg et al. [1] 논문에서 잘 학습된 BERT를 활용하여 각 문장별 representation을 통해 도메인 별 구분이 가능함을 보여 줌.
- BERT 대신에, 학습된 Baseline NMT 를 활용하도록 함.

[1] Aharoni et al. Unsupervised Domain Clusters in Pretrained Language Models , ACL2020

2.2 Features - Domain difficulty

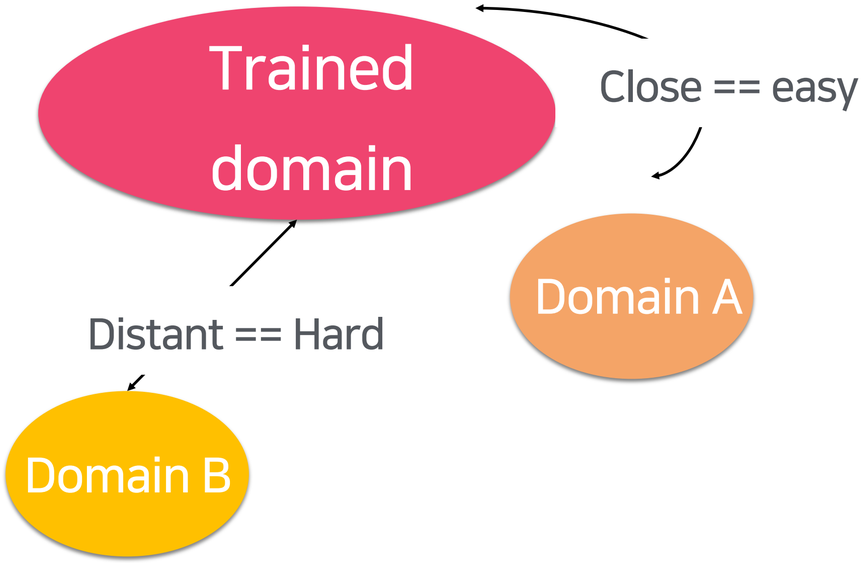
새로운 도메인의 난이도

방법 2) Baseline 모델이 해당 문장을 얼마나 잘 번역하는지 단일 언어 문장만 활용하여 추출

Intuition)

모델이 안 본 문장은 잘 생성 못하고 생성시 확신을 못 가짐

- Trained domain(seen domain) => Easy/Confident
- Similar domain => Relatively easy/confident (Domain A)
- Distant domain => Hard/Uncertain (Domain B)



"오늘은 더 나아지신 것 같습니다." - > Okay !

"코로나 바이러스는 기침과 고열을 동반하는 것으로 밝혀졌습니다." -> Hard !

2.2 Features - Domain difficulty

- Least-confidence score
  - $\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}} 1 - \mathrm{P}_{\theta}(\mathrm{y}_{\mathrm{i}}^{*}|\mathrm{x}, \mathrm{y}_{<\mathrm{i}}^{*})$
    - $\mathrm{P}_{\theta}(\mathrm{y}_{\mathrm{i}}^{*}|\mathrm{x}, \mathrm{y}_{<\mathrm{i}}^{*})$:decoding step의 i 번째 step에서의 확률 값
- Margin score
  - $-\frac{1}{m}\sum_{i=1}^{m} P_{\theta}(y_{i,1}^{*}|x, y_{<i}^{*}) - P_{\theta}(y_{i,2}^{*}|x, y_{<i}^{*})$
- Average token entropy
  - $\frac{1}{m}\sum_{i=1}^{m} entropy(P_{\theta}(y_{i}^{*}|x, y_{<i}^{*}))$
- LaBSE[2] score
  - LaBSE를 활용한 baseline model이 생성한 문장과 원래 입력 문장 간의 의미론적 유사도 측정
- 모든 feature 들은 monolingual 문장 에서 측정 가능
- 모델이 얼마나 확신을 가지면서 잘 생성했는지 측정

[2] Feng et al. Language-agnostic BERT Sentence Embedding.

2.2 Features – Corpus-level

Finetuning 하는 in-domain 데이터 안의 정보량

방법) in-domain 데이터가 이전에 사용한 general dataset에서 보지 않은 정보를 가지고 있는지 두 training datasets 간의 유사성 측정

Training dataset level에서 정보량을 측정 => Corpus-level features

- Baseline NMT 학습 문장과 유사한 training data로만 이뤄져 있으면 → Less informatic
- Baseline NMT 학습 문장과 상이한 training data로만 이뤄져 있으면 → More informatic

2.2 Features – Corpus-level

DEVIEW 2021

Corpus-level features

- A mount of tokens in S_d^s , where S_d^s is the monolingual sentences of the in-domain
- The average sentence length in tokens/characters
- Type-token ratio
- Vocabulary overlap ratio between the general dataset and S_d^s
- The number of unique tokens in S_d^s
- The number of sentences in S_d^s

- 모든 corpus-level feature 는 monolingual sentence로 부터 추출됨

10k training samples

Number of tokens : 10000

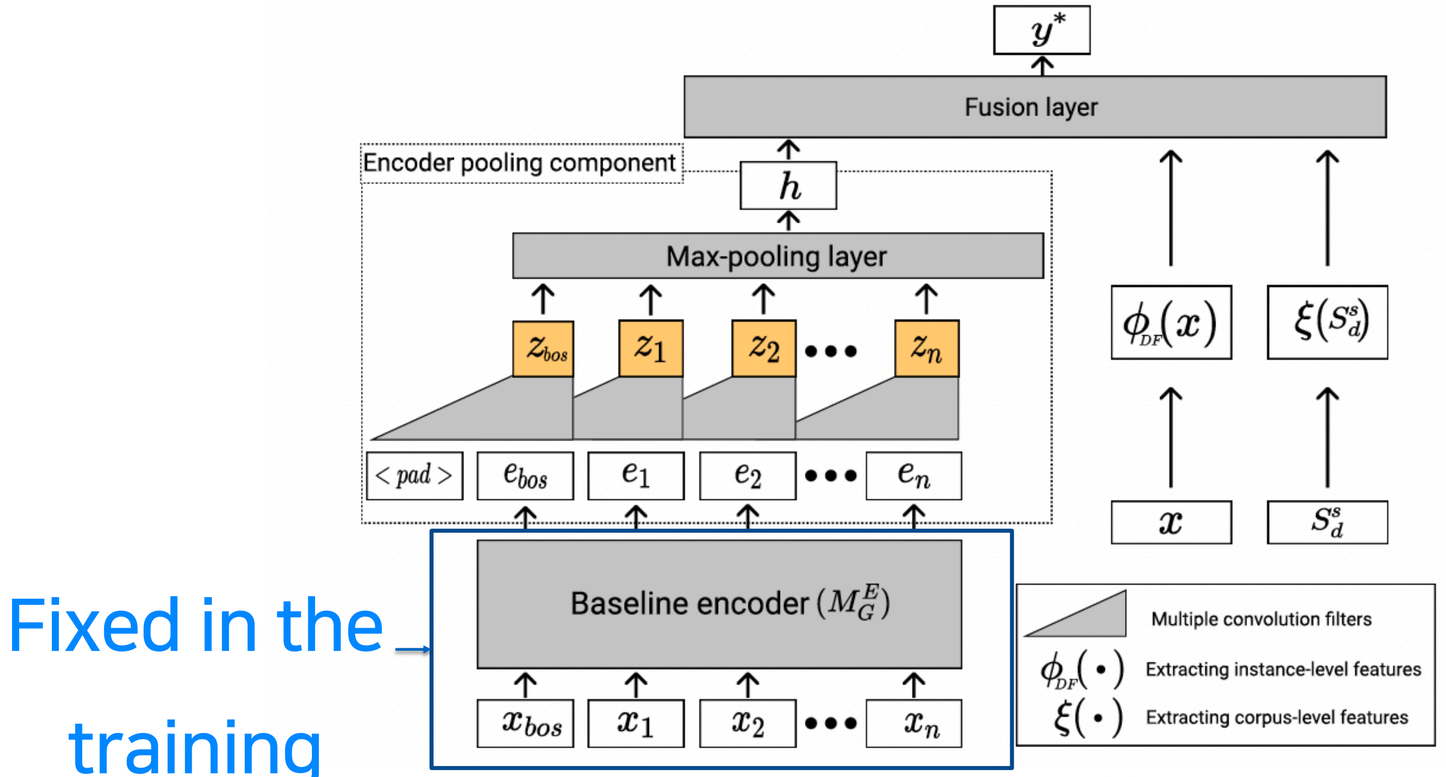
Vocabulary overlap : 0.5

⋮

2.3 Overall Architecture

Encoder pooling components [2] : 도메인 특성 추출

Fusion Layer: 추출된 features와 encoder pooling 의 output 조합 후 예측



Fixed in the training

[2] Kim et al. Convolutional Neural Networks for Sentence Classification. EMNLP 2014.

2.4 Summary of methods

Domain adaptation시에 영향을 줄 것으로 예상 되는 요인:

- 문장이 가지고 있는 도메인 특성
  - Baseline NMT encoder representation 사용
- 새로운 도메인의 난이도
  - Domain difficulty features의 정의 및 활용
- Finetuning 하는 in-domain 데이터 안의 정보량
  - Corpus-level의 features로 정보량 추출

2.4 Summary

다음의 세가지 조건을 만족시켜야 함

1) 적은 training anchor point를 사용
   -> Instance-level framework 를 제안하여 해결

2) 병렬 문장 없이 단일 언어 문장을 활용하여 모델이 예측
   -> 모든 features는 monolingual sentence만으로도 구할 수 있음

3) 임의의 도메인에도 정확한 성능
   -> Domain difficulty features와 NMT encoder representation을 활용

3.Expereiments and Results

3.1 Experiment settings

NMT model

Baseline model : Transformer 구조와 WMT-20 (en-de,de-en)데이터를 활용하여 학습

Finetuning model : Baseline model 을 추출한 In-domain corpus를 활용해서 finetuning

Data

5개의 도메인을 활용함 (Koran, medical, Law, subtitles, IT) [1]

Anchor points : 0k, 1k , 10k ,20k , 100k

- Koran과 같이 전체 데이터의 개수가 100k 아래인 경우 20k 데이터 까지만 만듦
- Leave-one-out setting : 4개의 도메인의 anchor point들로 학습하고 한개의 도메인으로 평가

Label(y)

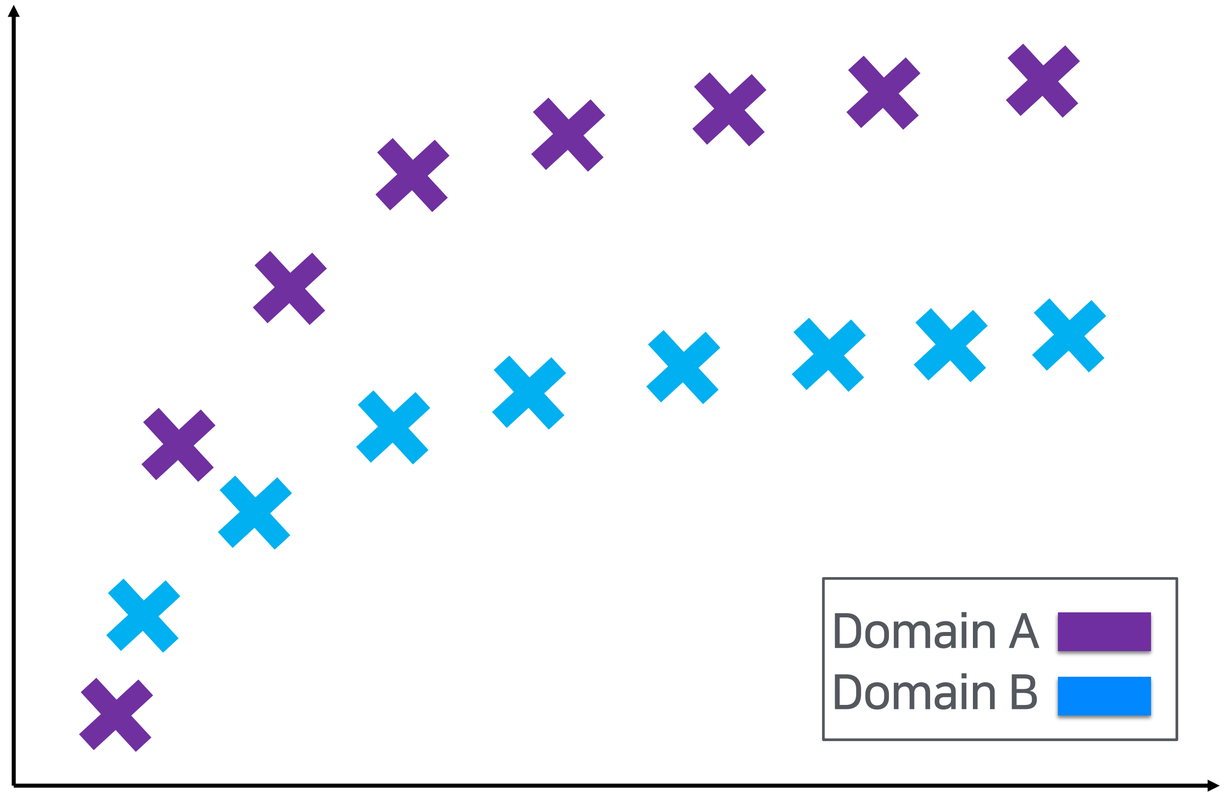
- 문장 레벨에서는 BLEU 보다 chrF 값이 더 실제 사람 평가와 correlation이 강해서 chrF값을 y값으로 사용
- 관심을 가지는 Corpus-level의 값은 각 문장의 chrF 값의 평균 값으로 정의
- 각 y 값은 실제로 finetuning후에 얻어낸 chrF 결과 값을 사용하도록 함.

[1] Aharoni et al. Unsupervised Domain Clusters in Pretrained Language Models , ACL2020

3.2 Baselines

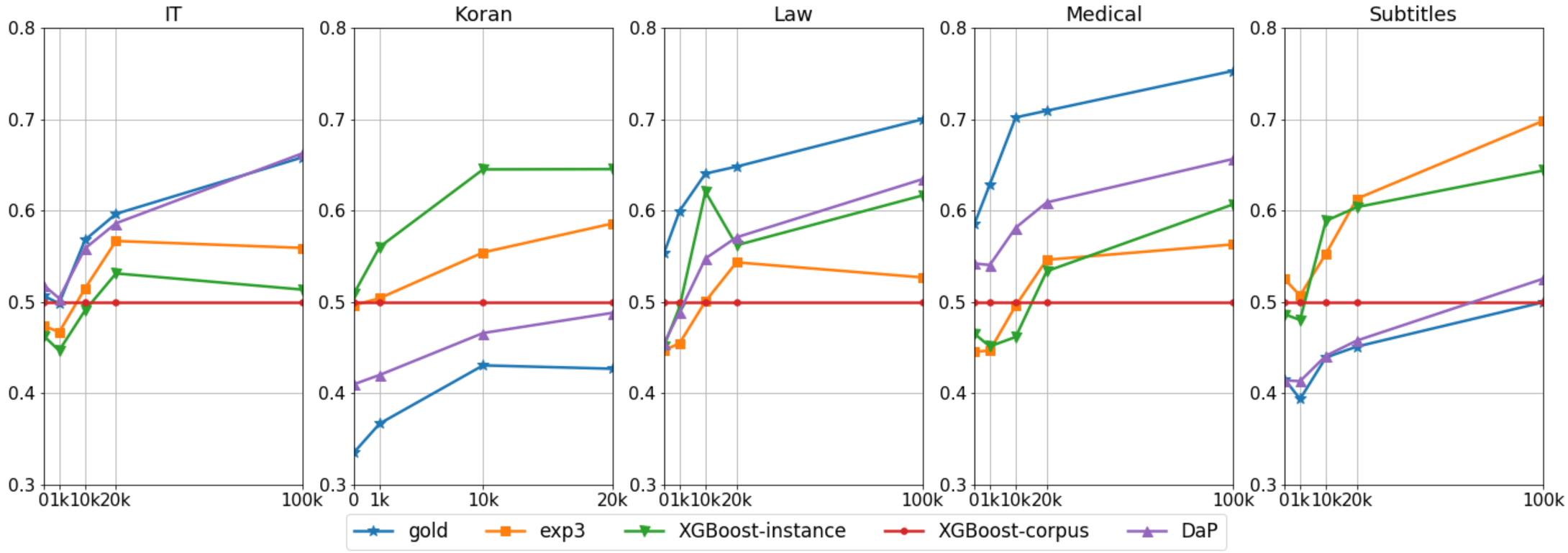
- Exp3
  - 19개의 training anchor points를 활용해서 Exp3 함수를 fitting (exp3: $y = c - e^{(-ax+b)}$)
- XGboost
  - Tree 갯수 : 100개 , Maximum dept 10 .
  - Xgboost-corpus : Corpus-level frame work 버전의 XGboost 모델
  - Xgboost-instance: Instance-level framework 버전의 XGboost 모델
    - Domain difficulty features와 corpus-level features 만 사용
- DaP/{features_name}
  - DaP에서 {features_name}을 뺀 모델
  - DaP/DF : Domain difficulty features들을 제거
  - DaP/NMT Enc : Encoder features를 제거
  - DaP/Corpus : Corpus-level features를 제거

3.2 Performance on the unseen domain



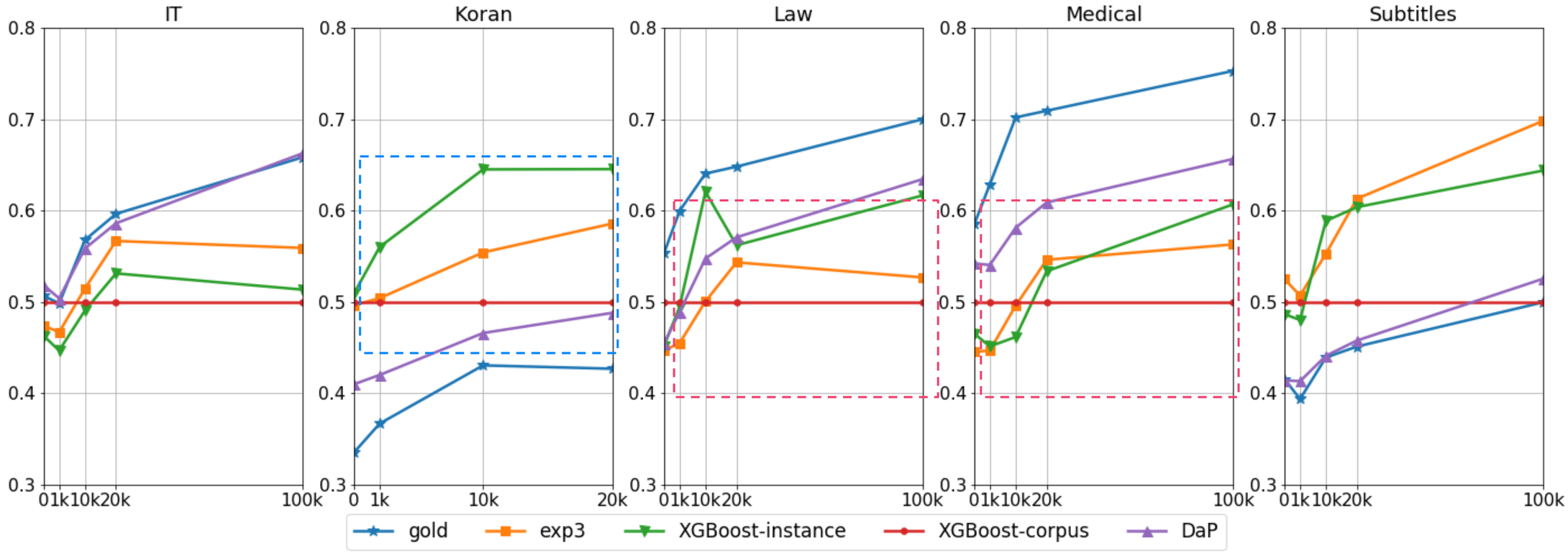
Domain C의 성능 그래프는
어떻게 그려질까?

3.2 Performance on the unseen domain



* x-axis : 데이터 갯수 , y-axis : corpus chrF 값

3.2 Performance on the unseen domain



Overestimation problem

Underestimation problem

* x-axis : 데이터 갯수 , y-axis : corpus chrF 값

3.2 Performance on the unseen domain

	IT	Koran	Law	Medical	Subtitles	Avg
	German-English/FT					
exp3	0.125	0.292	0.305	0.397	0.321	0.288
XGboost-corpus	0.197	0.235	0.305	0.435	0.157	0.266
XGboost-instance	0.084	0.201	0.062	0.176	0.126	0.130
DaP	**0.009**	**0.058**	**0.057**	0.094	**0.015**	**0.047**
DaP / DF	0.011	0.065	0.058	0.117	0.022	0.055
DaP / corpus	0.049	0.045	0.097	0.117	0.052	0.072
DaP / NMTEnc	0.025	0.148	0.085	**0.081**	0.061	0.080
	English-German/FT					
exp3	**0.035**	0.18	0.116	0.114	0.112	0.112
XGboost-corpus	0.111	**0.081**	0.169	0.169	**0.029**	0.112
XGboost-instance	0.072	0.157	0.159	0.116	0.09	0.119
DaP	0.048	0.107	**0.123**	0.041	0.057	0.075
DaP / DF	0.065	0.102	**0.123**	0.044	**0.053**	0.077
DaP / corpus	0.048	0.086	0.126	0.043	0.063	**0.073**
DaP / NMTEnc	0.043	0.169	0.125	**0.016**	0.095	0.09

Corpus-level

Instance-level

Corpus-level

Instance-level

Instance vs Corpus

Instance-level 로 하는 모델이

전반적으로 아주 좋은 성능을 보여줌

* 각 결과값은 실제와 예측 값사이의 Root Mean Square Error(RMSE)를 의미

4.2 Performance on the unseen domain

	IT	Koran	Law	Medical	Subtitles	Avg
	German-English/FT					
exp3	0.125	0.292	0.305	0.397	0.321	0.288
XGboost-corpus	0.197	0.235	0.305	0.435	0.157	0.266
XGboost-instance	0.084	0.201	0.062	0.176	0.126	0.130
DaP	**0.009**	**0.058**	**0.057**	0.094	**0.015**	**0.047**
DaP / DF	0.011	0.065	0.058	0.117	0.022	0.055
DaP / corpus	0.049	0.045	0.097	0.117	0.052	0.072
DaP / NMTEnc	0.025	0.148	0.085	**0.081**	0.061	0.080
	English-German/FT					
exp3	**0.035**	0.18	0.116	0.114	0.112	0.112
XGboost-corpus	0.111	**0.081**	0.169	0.169	**0.029**	0.112
XGboost-instance	0.072	0.157	0.159	0.116	0.09	0.119
DaP	0.048	0.107	**0.123**	0.041	0.057	0.075
DaP / DF	0.065	0.102	**0.123**	0.044	**0.053**	0.077
DaP / corpus	0.048	0.086	0.126	0.043	0.063	**0.073**
DaP / NMTEnc	0.043	0.169	0.125	**0.016**	0.095	0.09

Ablation from full model

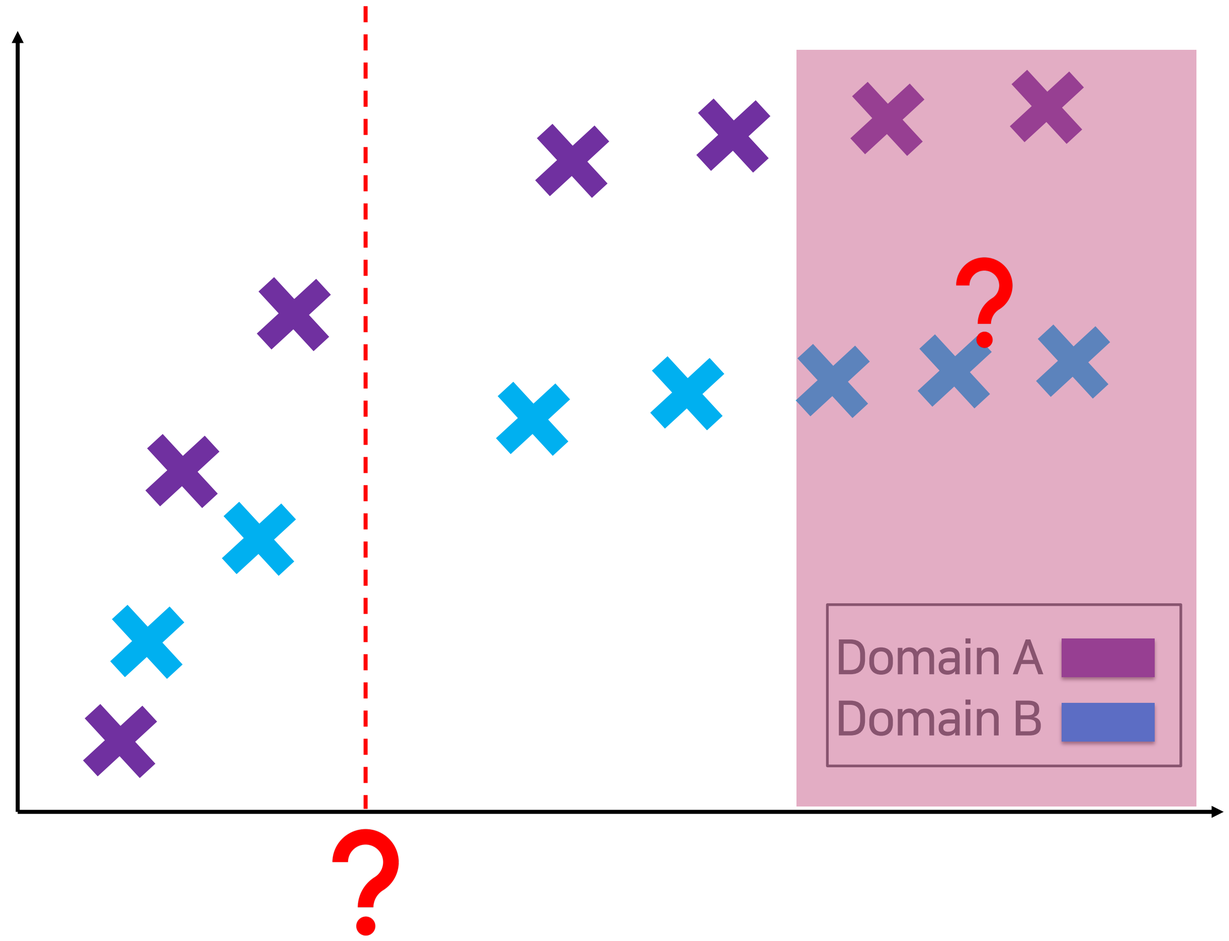
Ablation from full model

Ablation study

- DaP 가 [5/10] 의 case로 좋은 성능을 보여줌
- NMT Encoder representation이 성능 예측에 중요한 역할을 하는 것으로 볼 수 있음

* 각 결과값은 실제와 예측 값사이의 Root Mean Square Error(RMSE)를 의미

3.3 Interpolation and Extrapolation



Domain C의 성능 그래프는
어떻게 그려질까?

3.3 Interpolation and Extrapolation

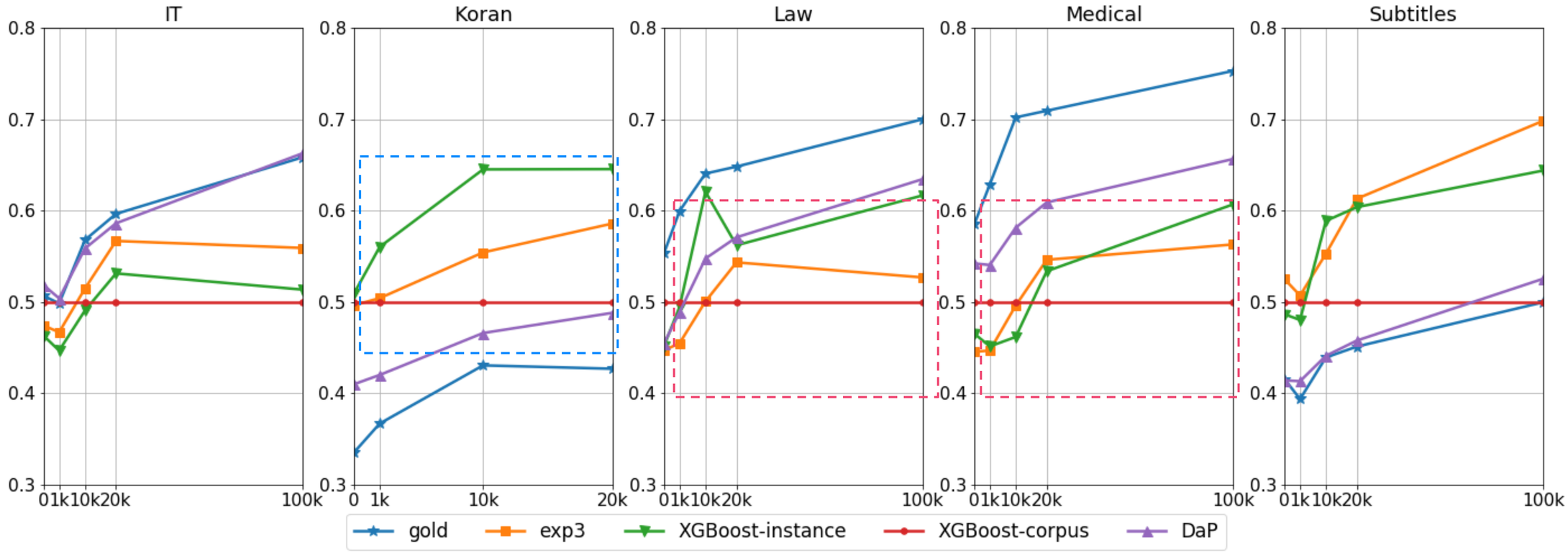
Interpolation | Extrapolation

model	3K	40k	160k
exp_3	0.1087	0.1640	0.1934
XGboost-instance	0.1546	0.0922	0.1125
DaP	**0.0063**	**0.0080**	**0.0413**

- Training 시에 본 anchor point는 (1k,10k,20k ,100k)
- Interpolation 결과 : 3k, 40k ; Extrapolation 결과 160k
- Extrapolation이 가장 어렵기 때문에 성능이 전반적으로 저하되지만 여전이 좋은 성능을 보여줌

* 각 결과값은 실제와 예측 값사이의 Root Mean Square Error(RMSE)를 의미

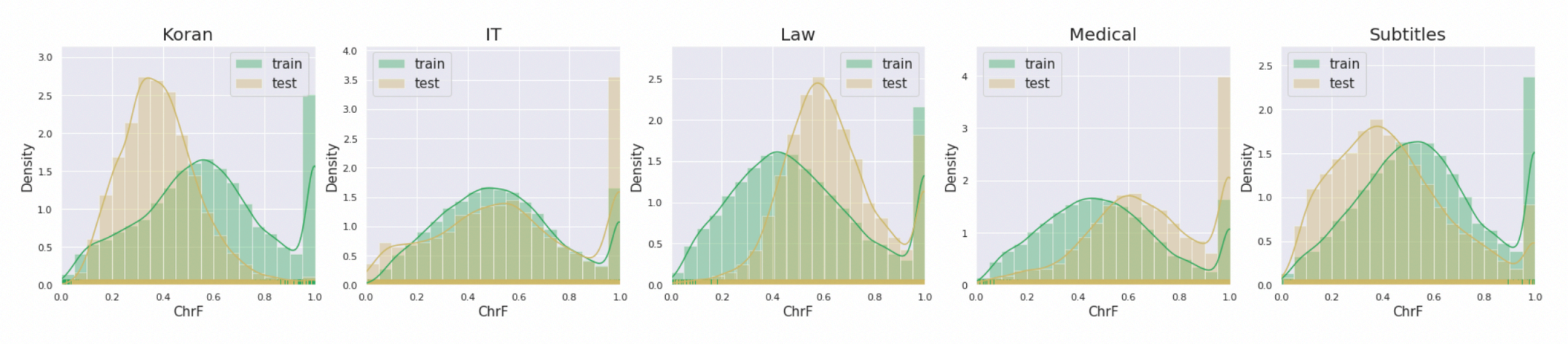
3.4 Analysis of errors



Overestimation problem

Underestimation problem

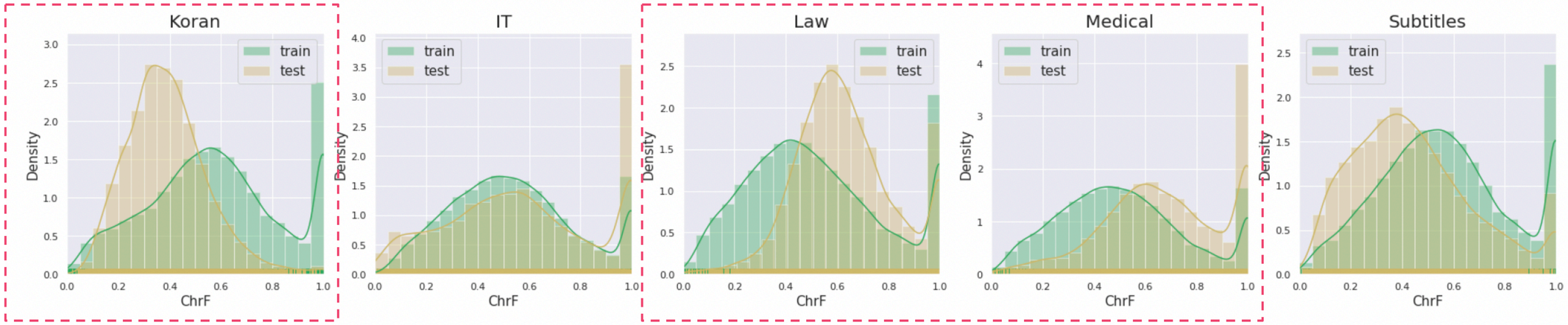
3.4 Analysis of errors



각 도메인 마다의 leave-one-out setting에서 train/test 데이터 chrF 분포 시각화

3.4 Analysis of errors

Highest shift

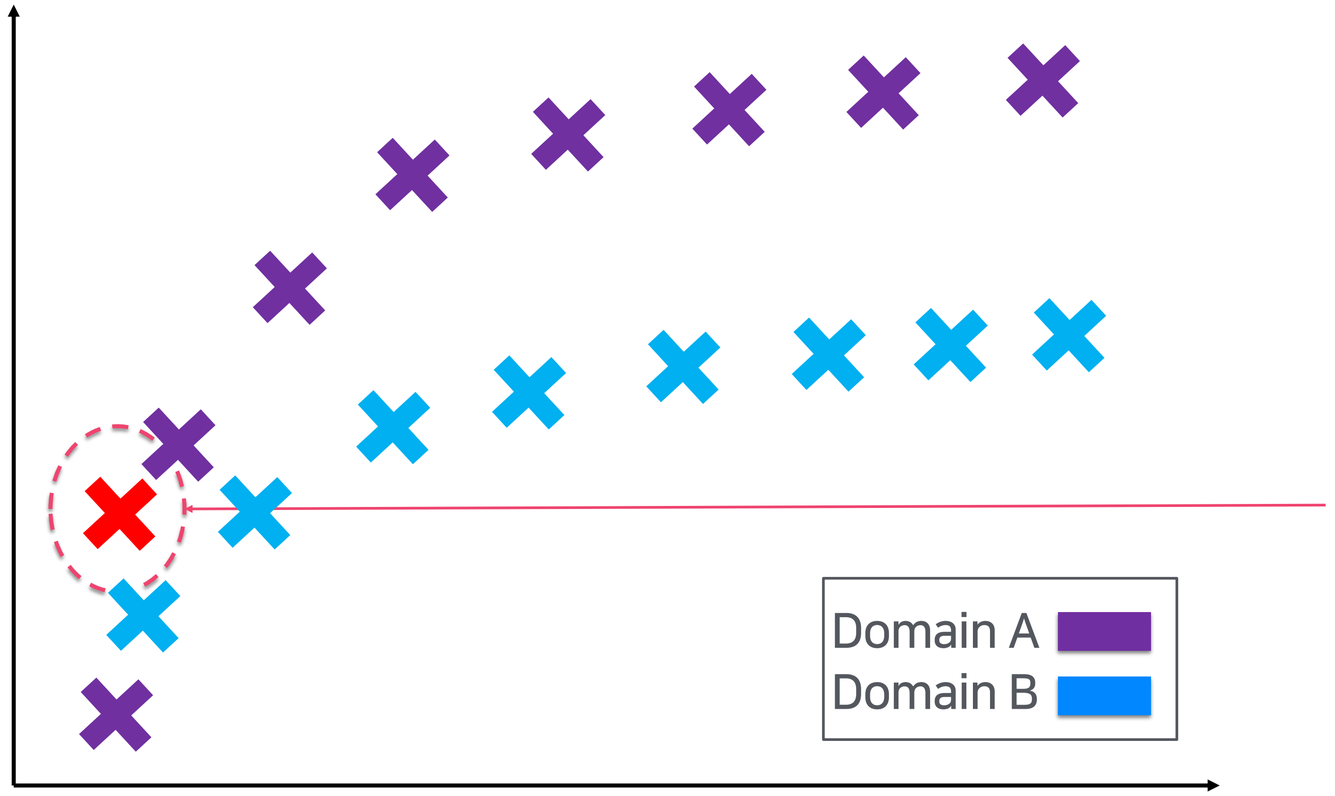


각 leave-one-out setting에서 train/test 데이터 chrF distribution 분포 시각화

Train 과 Test 분포가 큰 차이를 가지는

leave-one-out setting에서 큰 에러를 가짐

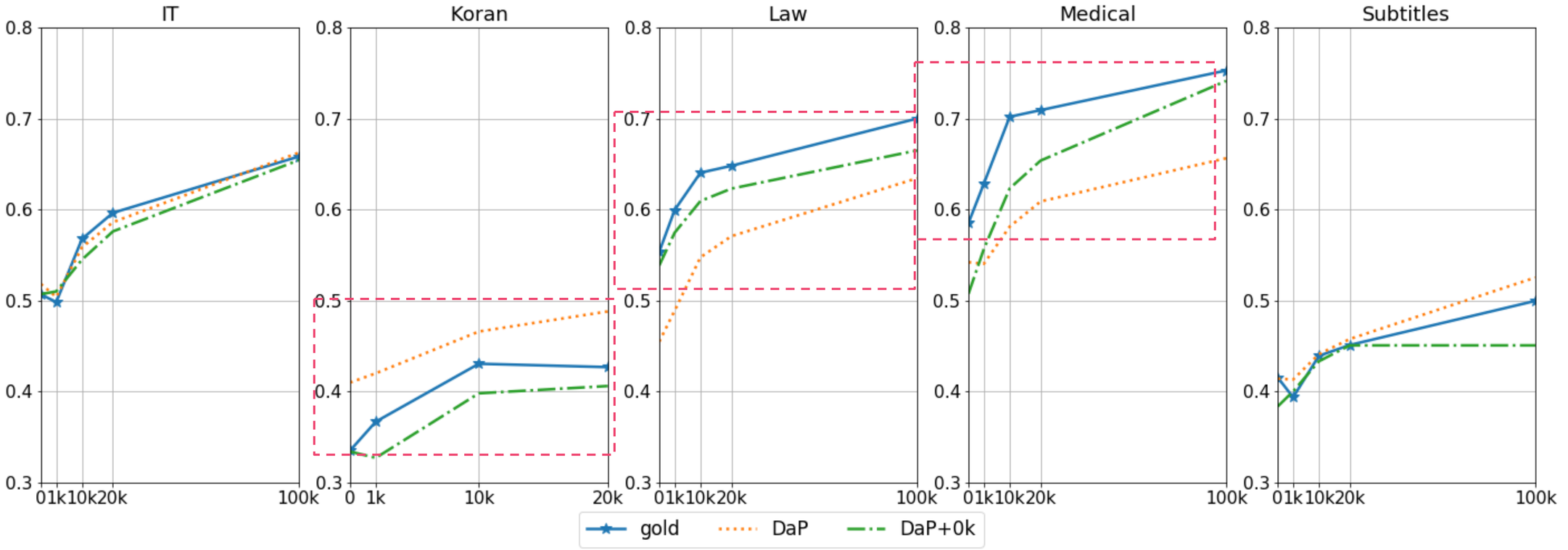
3.5 Performance with 0k anchor point



Domain C의 성능 그래프는
어떻게 그려질까?

보지 않은(Unseen domain)의
0K anchor point를 준다면?

3.5 Performance with 0k anchor point



전반적인 overestimation/underestimation 문제가 개선됨

* 각 결과값은 실제와 예측 값사이의 Root Mean Square Error(RMSE)를 의미

3.6 Performance on Ko-En pair

	Colloquial	News	Culture	Government	Ordinance	Avg
	Korean-English/FT					
exp3	0.039	**0.042**	0.043	0.124	0.192	0.088
XGboost-cropus	0.085	0.136	0.094	0.151	0.209	0.135
XGboost-Instance	0.213	0.217	0.062	0.078	0.122	0.138
DaP	**0.019**	0.103	**0.025**	0.103	0.096	**0.076**
DaP / DF	0.025	0.120	0.029	0.083	0.098	0.078
DaP / corpus	0.037	0.135	0.031	**0.074**	0.104	0.078
DaP / NMTEnc	0.025	0.286	0.026	0.082	**0.083**	0.112

전반적으로 DaP 모델이 Ko-En 데이터에 대해서도 우수한 성능을 보이고 있음을 알 수 있음

* 각 결과값은 실제와 예측 값사이의 Root Mean Square Error(RMSE)를 의미

4. Conclusion & Future work

DEVIEW 2021

Take-home messages

- Instance-framework는 기존의 corpus-level framework보다 적은 데이터 만으로 성능 예측을 정확하게 할 수 있게 해 준다.
- Domain adaptation 성능에 영향을 줄 수 있는 features와 그 모델링 technique들
    - NMT enc representation의 도메인 특성 capture 효과

Future work

- 실제 데이터에 적용
- Active learning 으로의 확장
- 다양한 DA 알고리즘에 적용 가능한지 확인

Thank You