Understanding Back-Translation at Scale

Sergey Edunov, Myle Ott, Michael Auli, David Grangier

EMNLP 2018

오지은

index

introduction

introduction

- NMT를 위해서는 (아주) 많은 paired sentence가 필요하지만, 사람이 만든 paired data의 양은 한정되어 있다
- 반면 monolingual data는 언제나 비교적 풍부하다 (위키피디아를 비롯한 모든 웹페이지들)
- 그럼 번역 문제에도 monolingual data를 이용할 수 있으면 좋지 않을까?
- **Back-translation**: Auxiliary translation system을 따로 학습해서, target 언어로부터 source 언어로 번역된 문장(**synthetic data**)을 생성하는 것
  - 이 인위적으로 만들어진 데이터를 원래의 parallel data에 섞어서 학습에 사용
  - source 문장은 인공 데이터, target 문장은 진짜 bitext의 타겟 언어 문장이 됨
  - 즉 {I am a student:Je suis etudiant}에서 je suis etudiant를 번역하여 {I am student:Je suis etudiant} 데이터를 인위적으로 생성하여 학습 데이터에 집어넣음

Back-translation

- Understanding Back-Translation **at Scale**: 원래의 bitext에 아주 많은(hundreds of millions) synthetic data를 추가함으로써 back-translation에 관하여 **큰 스케일로** 알아보도록 하겠다
- synthetic data를:
  - 어떻게 만들어야 품질이 좋은지
  - 어느 정도 학습하는지
  - 만들 때 데이터의 도메인이 성능에 어떤 영향을 미치는지
  - 학습에 사용할 때 진짜 데이터와의 비율은 어느 정도가 좋은지

본론: back-translation

generating synthetic sources

- back-translation에는 두 가지 방법이 사용됨:
  - beam search
  - greedy search
  - 두 방법 모두 maximum a-posteriori (MAP) 방식으로, 주어진 입력에 대해 가장 확률이 높은 문장을 구하는 것
- 그런데 이 MAP 방법을 쓰면 생성되는 번역 문장이 별로 풍부하지 못한데, 모호성이 높은 상황에서 언제나 most-likely한 경우를 고르기 때문
- 그렇기 때문에

  ① sampling

  ② beam search에 노이즈 추가하기

  를 고안해보겠다

generating synthetic sources

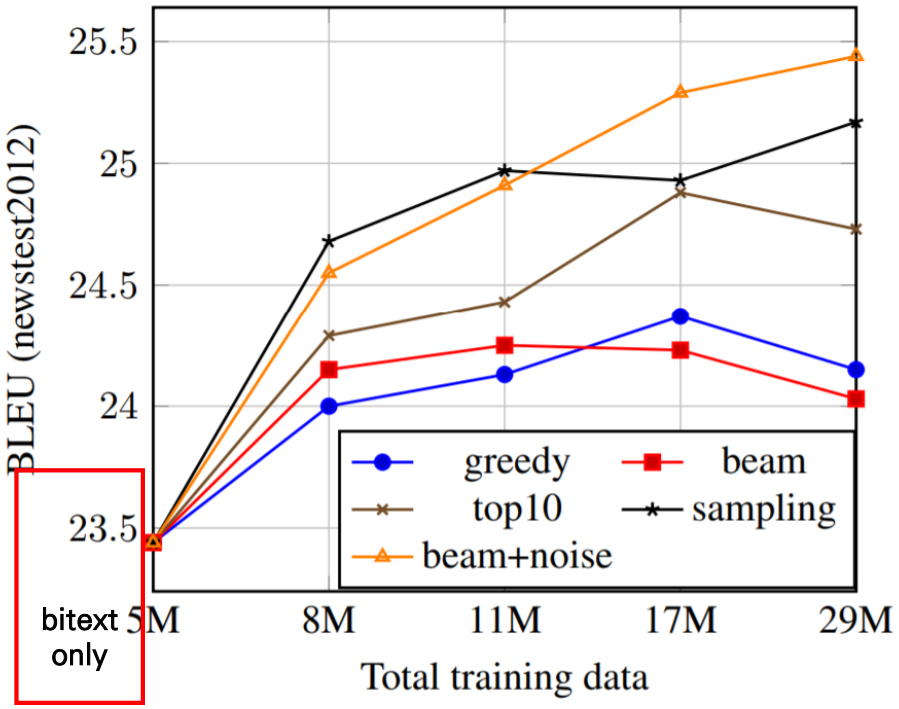
	Reference	사람 과 컴퓨터 가 의사소통 을 합니다.
MAP	Beam Search	사람과 컴퓨터 가 의사소통 을 하기 합니다.
non MAP	Random Sampling	역사 사람 은 과 전산 입니다 입니다 자동차 기술
non MAP	Top-10 Sampling	사람 은 컴퓨터 는 기술 및 통신 이다.
MAP	Beam + Noise	과 〈BLANK〉 컴퓨터 가 의사소통 〈BLANK〉 합니다.

- greedy decoding, beam search: MAP 방식. 가장 확률이 높은 단어를 고름 (beam search는 greedy의 발전 버전)
- random sampling: 모델의 확률 분포에 따라 랜덤하게 단어 선택 (가끔 말도 안 되는 단어를 선택할 수 있음)
- restricted (Top10) sampling: 확률이 가장 높은 top 10개 단어 중에서 하나를 고름. random sampling보다는 덜 랜덤함
- beam+noise: 입력 문장에 노이즈(delete, replace, permutate)를 적용하고 beam search 사용

generating synthetic sources

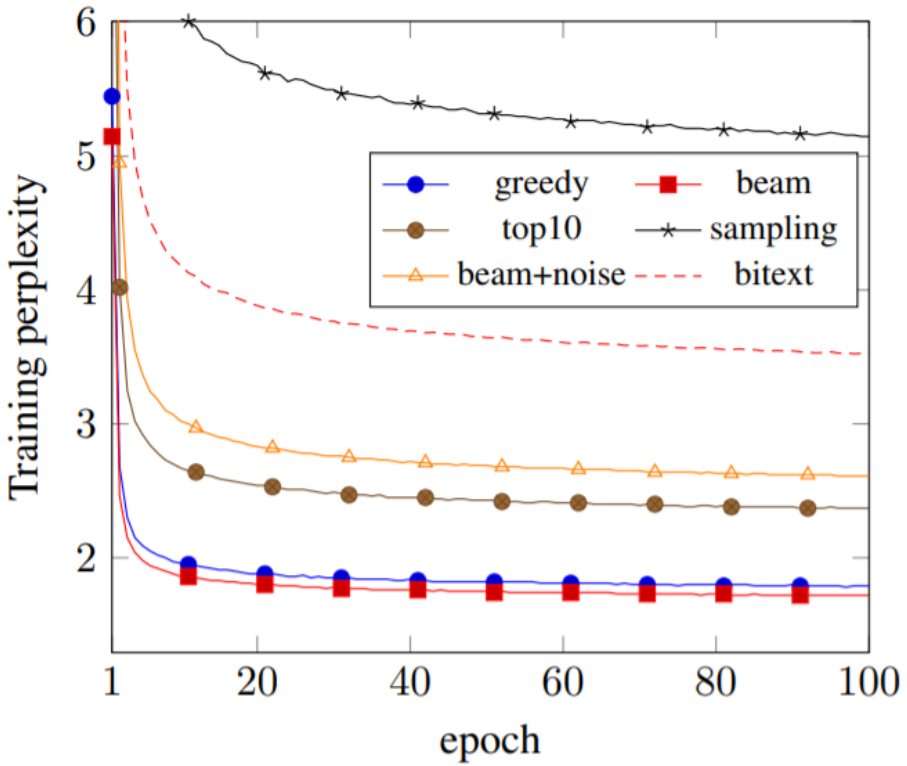
	news2013	news2014	news2015	news2016	news2017	Average
bitext	27.84	30.88	31.82	34.98	29.46	31.00
+ beam	27.82	32.33	32.20	35.43	31.11	31.78
+ greedy	27.67	32.55	32.57	35.74	31.25	31.96
+ top10	28.25	33.94	34.00	36.45	32.08	32.94
+ sampling	28.81	34.46	34.87	37.08	32.35	33.51
+ beam+noise	29.28	33.53	33.79	37.89	32.66	33.43

bitext 5백만 문장에 대해 학습한 후 인공 데이터 2천4백만 문장을 더하여 학습한 결과



- restricted sampling, sampling, beam+noise가 보통의 pure MAP 방식보다 좋음
- beam search는 항상 가장 그럴듯한 결과만 내기 때문에 만들어진 인공 데이터(즉 소스 문장)의 richness와 diversity를 해친다
- 그러나 noisy한 소스 문장을 주면 모델은 알맞은 번역을 만들기 위해 더 노력해야 한다 (일종의 DAE처럼 작용할 수 있다) → 더 강한 training signal을 줌

analysis of generation methods



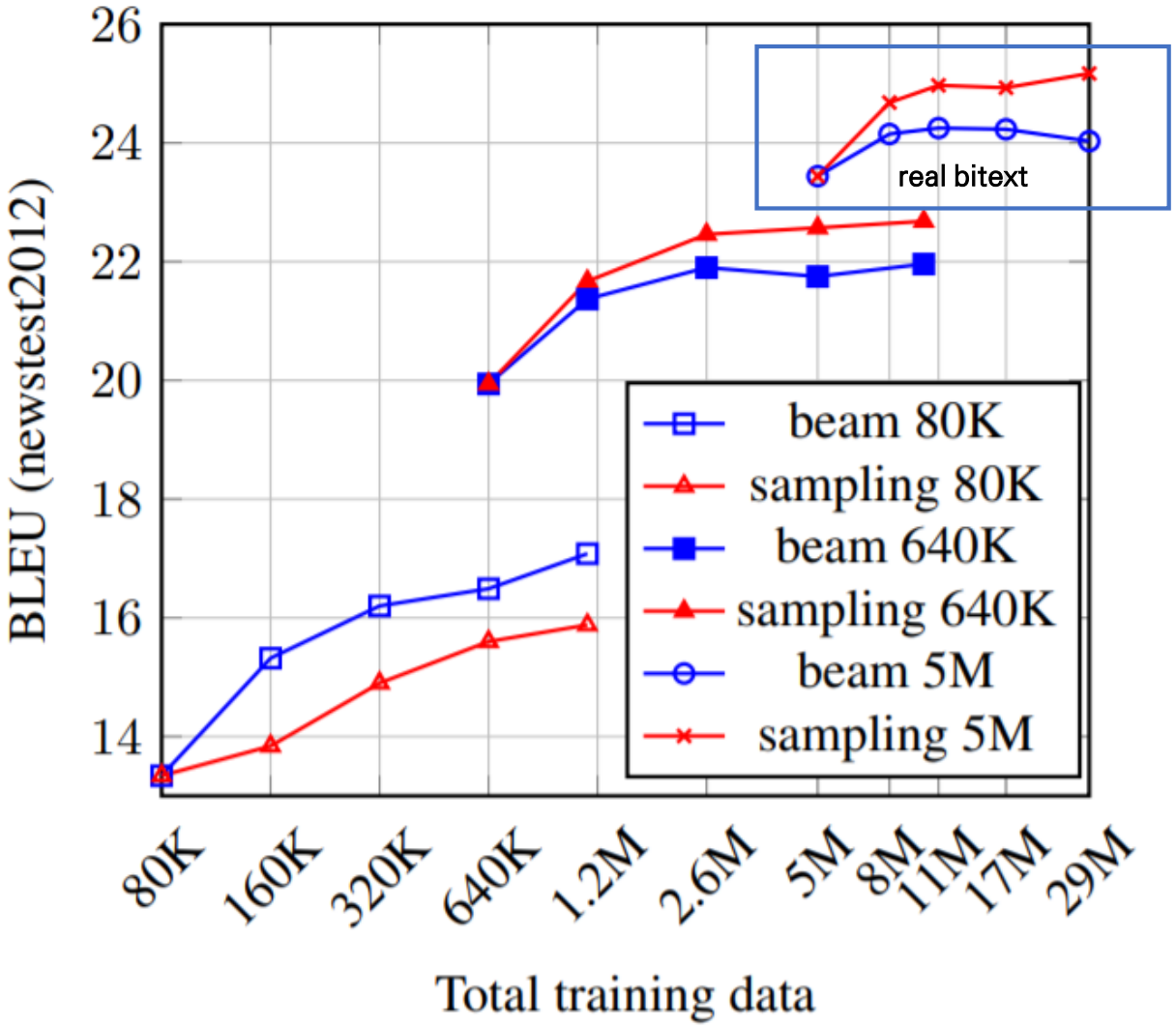
	Perplexity
human data	75.34
beam	72.42
sampling	500.17
top10	87.15
beam+noise	2823.73

- beam search로 만들어진 데이터가 가장 perplexity가 낮음 → 가장 'predictable' 함
- richness, variability가 떨어진다는 뜻
- 맞히기 어려운 **noisy 데이터가 오히려 학습에 더 도움을 줄 수 있다**

resources: low vs high

- 지금까지의 실험에는 많은 병렬 데이터를 사용했음 (back-translation 모델이 쓸 만한 성능을 냈기 때문에 학습 데이터로 사용할 수 있었음)
- 그만한 데이터가 없는 세팅에서는 back-translation 모델의 성능이 떨어질 수밖에 없다
- 그런 상황에서도 non-MAP 방법들이 유용할까?
- 그것을 알아보기 위하여 데이터 중에 일부를 골라 low-resource setting을 시뮬레이션하여 실험

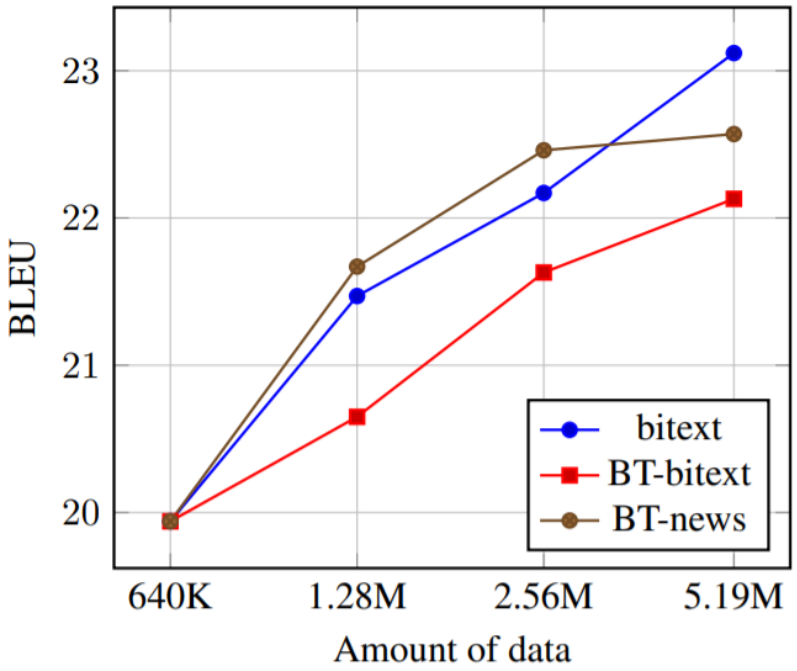
→ **resource-poor (80K) 실험에서는 sampling의 성능이 떨어지고 beam search는 안정적**



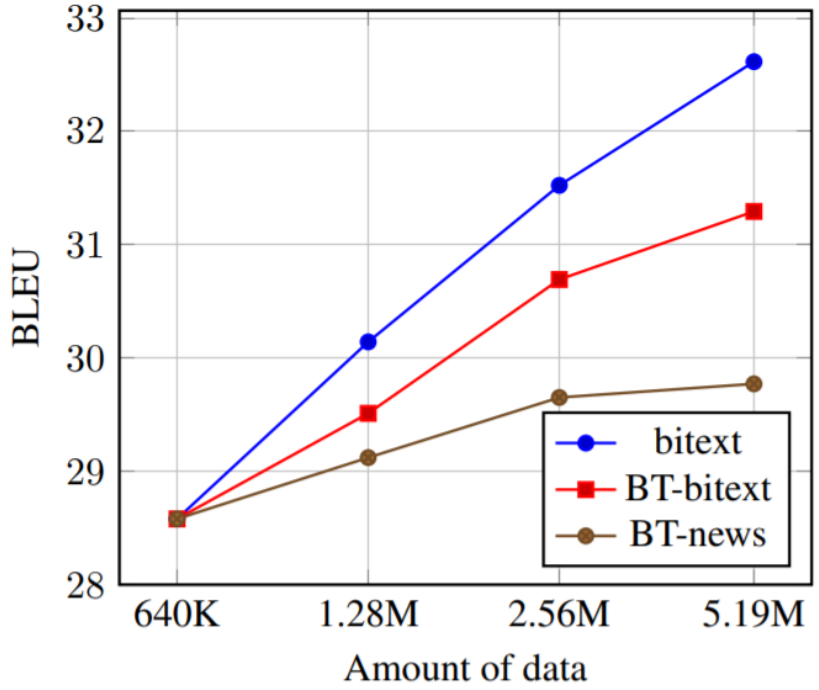
domain of synthetic data

- back-translation으로 synthetic data를 만드는 데 사용된 데이터의 도메인이 학습 최종 결과에 영향을 미칠까?
- 실제 bitext와 도메인에 따른 synthetic dat를 비교하기 위한 실험: 우선 bitext에서 640K 문장을 subsample 후
  - ① 나머지 bitext
  - ② 나머지 bitext의 back-translated 데이터
  - ③ news 데이터로 back-translate한 데이터
  - 1, 2는 같은 target side를 공유함. bitext는 뉴스의 비중이 적지만 BT-news는 순전한 뉴스임

domain of synthetic data



(a) newstest2012

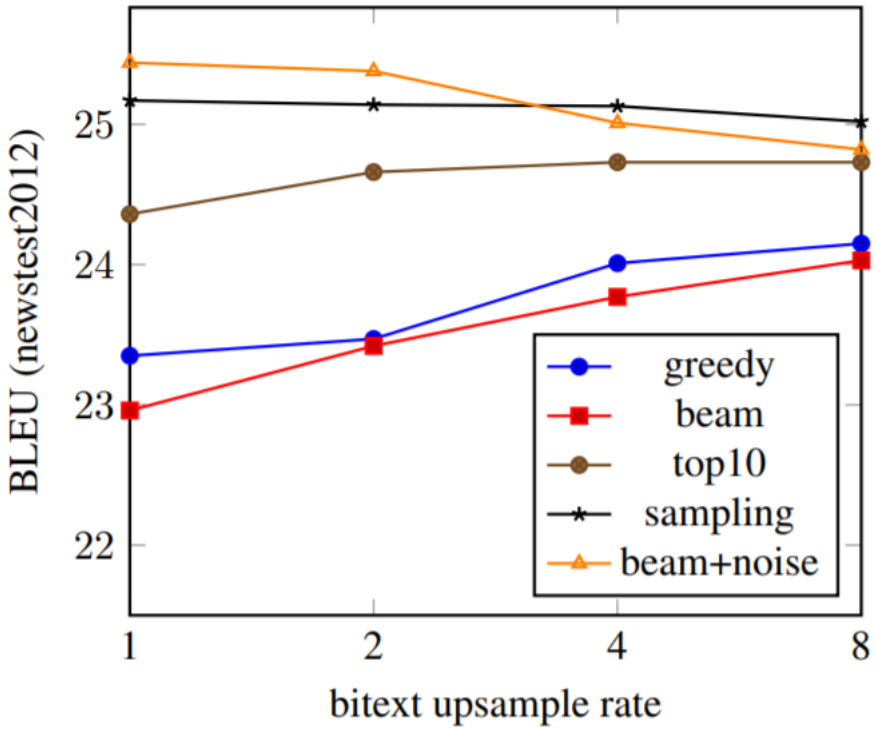


(b) valid-mixed

- (a): pure news로 테스트 / (b): 뉴스의 비중이 적은 혼합 데이터로 테스트
- 뉴스 데이터로 테스트할 때는 BT-news와 bitext의 성능 차이가 별로 나지 않음 → **도메인이 맞기만 하다면 synthetic data로 사람이 만든 데이터(bitext) 같은 효과를 낼 수 있다**
- 도메인이 달라지면 같은 도메인일 때만한 효과는 나지 않음. 그러나 back-translation을 쓰지 않은 것보다는 어쨌든 도움이 된다
- (b)의 경우를 볼 때, bitext 640K와 monolingual data 5M이 있다면 bitext 5M을 동원한 것과 꽤 유사한 효과를 낼 수 있음

upsampling the bitext

- 실제 학습 중에 bitext:synthetic data의 비율이 어느 정도여야 이상적인가?
  - ex: 데이터 안에 bitext가 5M 있고 synthetic이 10M 있고 upsampling rate=2라면 → 학습할 때 bitext를 synthetic의 2배 방문함
- beam, greedy 방법의 경우 rate가 높아야 좋음
- sampling, noise는 bitext를 많이 보지 않아도 됨
- sampling과 noise를 활용하면 synthetic data가 '학습하기 어려운' 데이터이기 때문에 더 강한 training signal을 제공 → bitext에 크게 의존하지 않음



large scale results

- back-translation으로 데이터를 늘려서 얻을 수 있는 최종 결과 탐구
- 영어-프랑스어: 35.7M 크기의 bitext에 다시 back-translation으로 얻은 데이터 31M을 더하여 번역을 학습함 (sampling 사용)
- 영어-독일어: 226M 크기의 back-translation data를 만들어 학습함 (upsample rate 16: bitext를 synthetic data보다 16배 많이 봄)

	news13	news14	news15
bitext	36.97	42.90	39.92
+sampling	**37.85**	**45.60**	**43.95**

	news13	news14	news15
bitext	35.30	41.03	38.31
+sampling	**36.13**	**43.84**	**40.91**

Conclusion

결론

- 데이터가 많으면 무조건 좋다
- 심지어 그 데이터가 다소 noisy하더라도 좋다
- 또는 데이터가 noisy하기 때문에 오히려 더 잘 학습된다
- 그러므로 **back-translation을 사용하면 거의 무조건 성능에 이득이다**
  - 이때 단순한 greedy decoding, beam search를 이용한 synthetic data도 좋지만
  - sampling, noising을 이용하면 학습할 때 DAE 효과를 주어 성능이 더 올라갈 수 있다
  - 단 low-resource 상황일 때는 sampling이나 noising이 생성하는 데이터의 품질이 너무 떨어지므로, 안정적인 pure MAP 방법이 더 유리할 수 있다

reference

- https://www.aclweb.org/anthology/D18-1045.pdf
- https://dev-sngwn.github.io/2020-01-07-back-translation/
- https://paperswithcode.com/sota/machine-translation-on-wmt2014-english-french

End of Document