Understanding Back-Translation at Scale

Sergey Edunov, Myle Ott,
Michael Auli, David Grangier (EMNLP2018)

紹介: 小町研 M1 勝又 智

概要

- NMT で tgt 側の monolingual data を使いたい！
  → back-translation (BT) によって擬似的な対訳データを作成, 学習データに追加する手法がよく用いられる

- この論文は BT の synthetic src について, いくつかの作り方を検討した

- Low-resource の設定で**なければ**, noisy synthetic src による学習データを使用した方が良いことがわかった
  ↑ noisy synthetic src の方が non-noisy より train signal が強いため, という主張

- 他にも, BT による synthetic data と genuine bitext を比較した

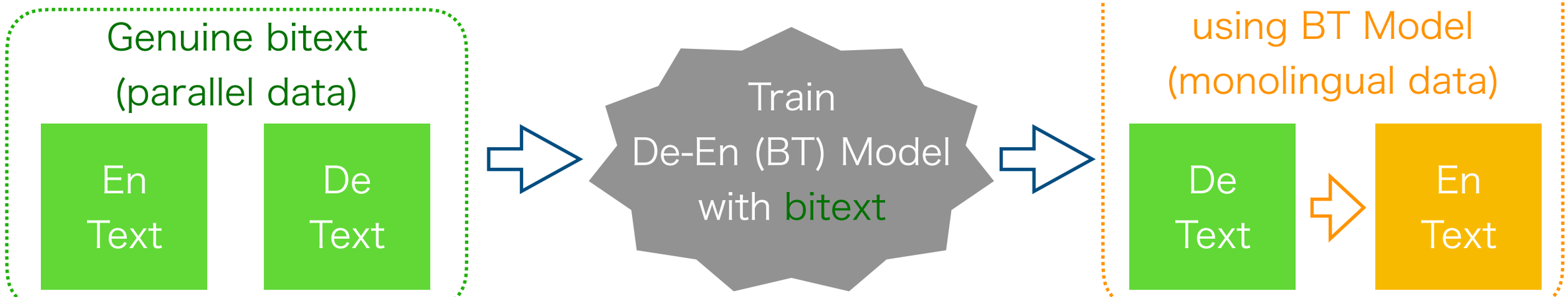
- なんか SOTA とった (WMT 2014 En-De, DeepL 比較)

概要

- NMT で tgt 側の monolingual data を使いたい！
  → back-translation (BT) によって擬似的な対訳データを作成, 学習データに追加する手法がよく用いられる

- この論文は **BT の synthetic src** について, いくつかの作り方を検討した

翻訳方向: English → German (自然なデータは緑, 擬似データは橙)



概要

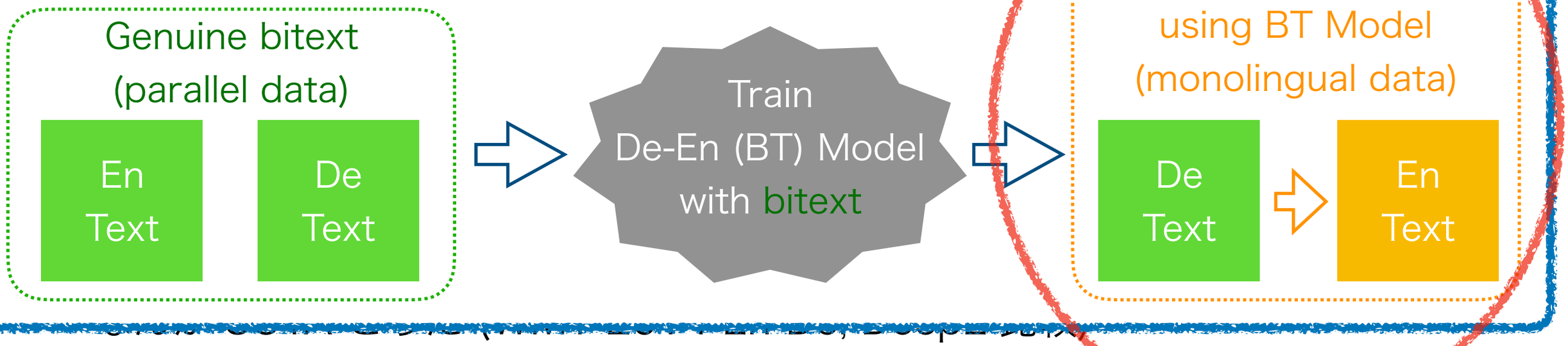
- NMT で tgt 側の mono
  → back-translation
  に追加する手法がよく

学習データ

この synthetic src を noisy になるよう
にした方がいい, というお話

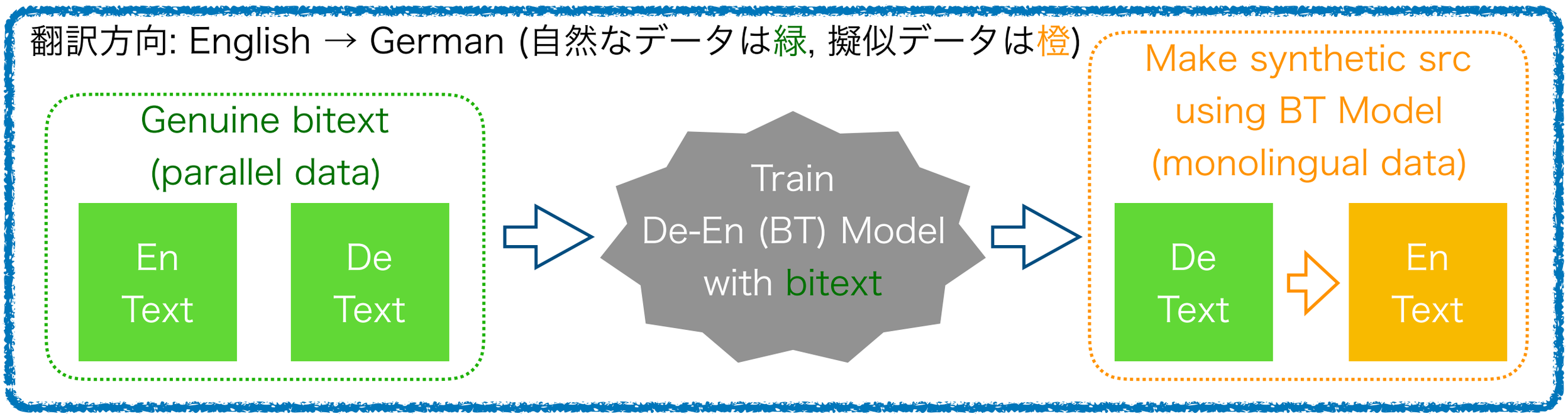
- この論文は **BT の synthetic src** について, いくつかの作り方を検討した

翻訳方向: English → German (自然なデータは緑, 擬似データは橙)



改めて
Back-Translation とは

- NMT で tgt 側の monolingual data を使いたい!
  → 逆方向の翻訳モデルを対訳データから学習,
  tgt の monolingual data に対して推論を行い, synthetic src を生成する.
  この synthetic data を学習データに加えて, 順方向のモデルを学習,
  性能向上を目指す. (Sennrich et al., ACL2016)
- 最近は Unsupervised MT でも使われたりしてる. (Lample et al., EMNLP2018)



改めて

Back-Translation とは

- NMT で **tgt 側の monolingual data** を使いたい!

  → 逆方向の翻訳モデルを対訳データから学習,

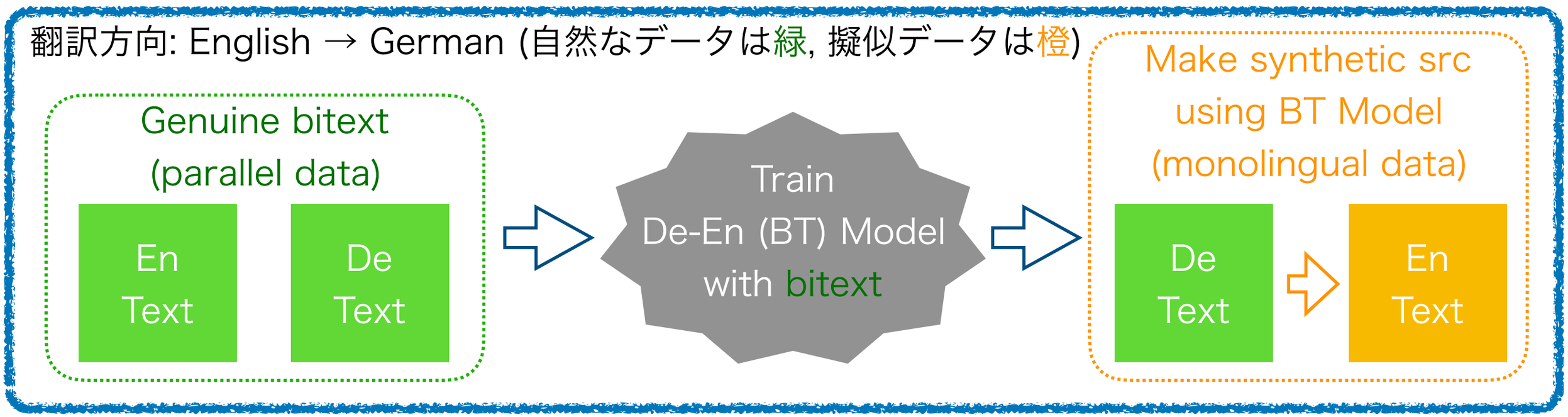
  tgt [illegible] src を生成する.

  こ[illegible] を学習,

  性[illegible]

Monolingual data は**人手**によるもの.
片側 (monolingual) だけなら適当にクロールすれば
データが得らるれので, このデータで何かできたら嬉しい

- 最近は Unsupervised MT でも使われたりしてる. (Lample et al., EMNLP2018)

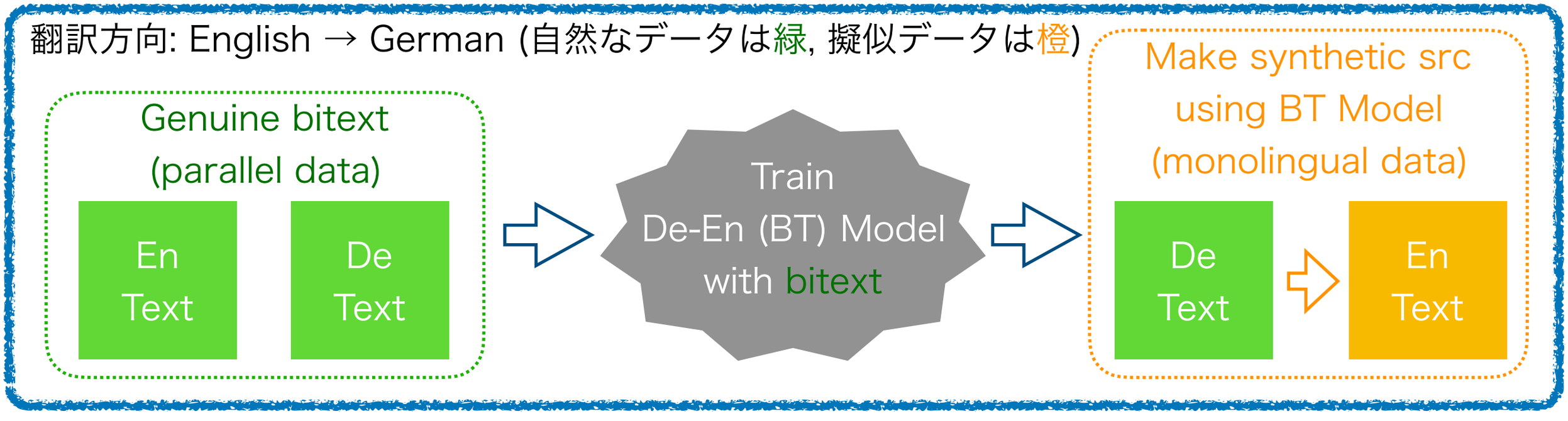


改めて

Back-Translation とは

- NMT で tgt 側の monolingual data を使いたい!
  → **逆方向の翻訳モデルを対訳データから学習,**
  tgt の monolingual data に対して推論を行い, synthetic src を生成する.
  この synthetic data を学習データに加えて, 順方向のモデルを学習,
  性能向上を目指す. (Sennrich et al., ACL2016)

- 最近は Unsupervised MT でも使われたりしてる. (Lample et al., EMNLP2018)



改めて

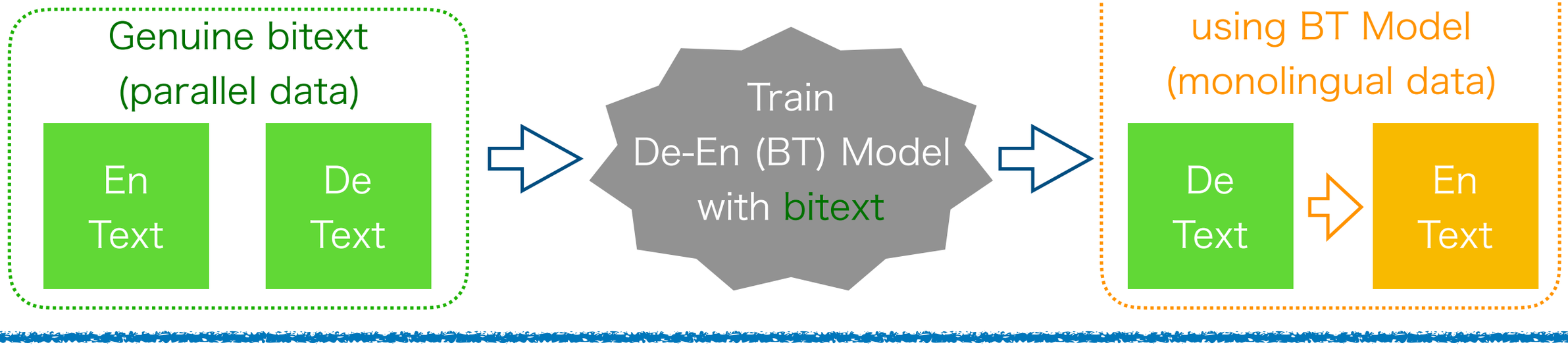
Back-Translation とは

- NMT で tgt 側の monolingual data を使いたい!
  → **逆方向の翻訳モデルを対訳データから学習,**
  tgt の monolingual data に対して推論を行い, synthetic src を生成する.
  この synthetic data を学習データ
  性能向上を目指す. (Sennrich et al.

この BT による synthetic src は
greedy or **beam search** で作るのが通例

- 最近は Unsupervised MT でも使われたりしてる. (Lample et al., MNLP2018)

翻訳方向: English → German (自然なデータは緑, 擬似データは橙)



改めて

Back-Translation とは

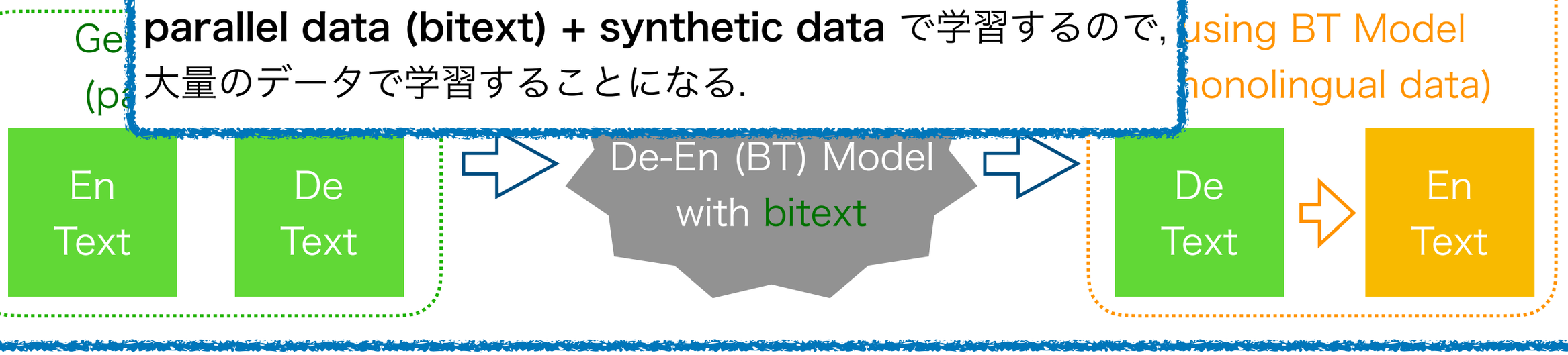
- NMT で tgt 側の monolingual data を使いたい!
  → 逆方向の翻訳モデルを対訳データから学習,
  tgt の monolingual data に対して推論を行い, synthetic src を生成する.
  この **synthetic data を学習データに加えて, 順方向のモデルを学習**,
  性能向上を目指す. (Sennrich et al., ACL2016)

- 最近 al., EMNLP2018)

Synthetic data = synthetic src + original tgt
（要するに synthetic data は擬似対訳データ）

最終的に順方向のモデルは
parallel data (bitext) + synthetic data で学習するので,
大量のデータで学習することになる.

翻訳方向:



Synthetic src の作り方に一工夫

- 一般に, BT は synthetic src 作成に greedy or beam search を行う
  → これらの decoding は最も確率の大きいもの (文, 単語) を出力としている
  (MAP 推定の出力の近似的なものとみなせる)

- 前提: MAP 推定に近似できる出力は rich translation になりづらい (Ott et al., ICML 2018)

- 仮説: greedy or beam search は出力分布の一番高いところしか見ていないので,
  生成される synthetic src は真のデータ分布を適切にカバーしていないのではないか?
  → synthetic src として regular なものを返すのではないか?

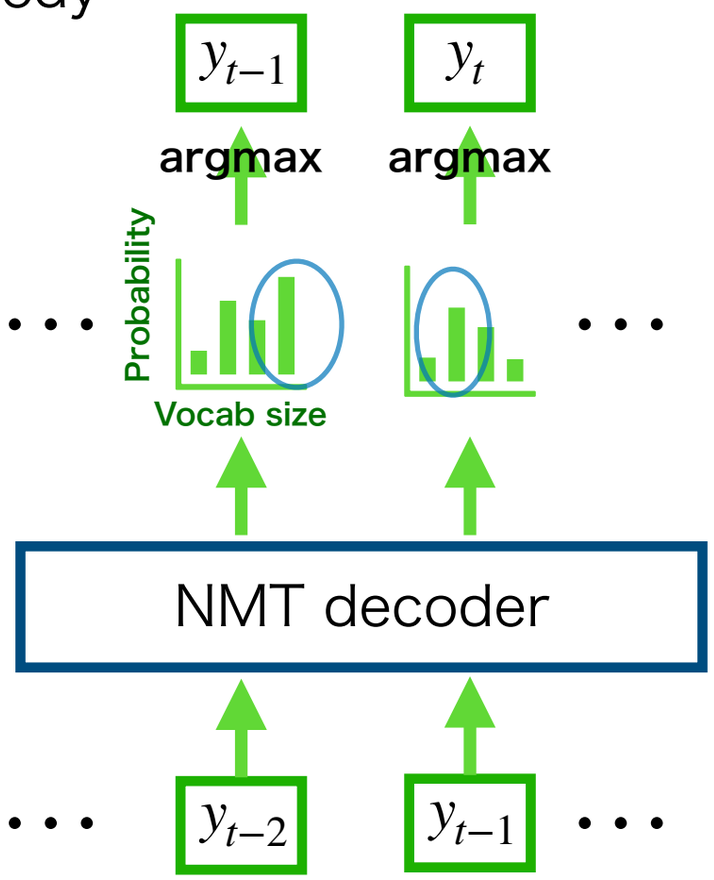
- 提案: 出力分布から sampling する or beam 出力に noise を乗せる

- 具体的にはイカの3種の手法を試した
  1. sampling: 出力分布に基づいて sampling (各ステップで)
  2. top-k: 出力分布のうち上位 k token をとり, renormalize, この分布から sampling
  3. beam+noise: beam 出力に対して, 人工的な noise (word drop, blank, shuffle) を乗せる

Synthetic src の作り方に一工夫

- 一般に, BT は synthetic src 作成に greedy or beam search を行う
  → **これらの decoding は最も確率の大きいもの (文, 単語) を出力としている (MAP 推定の出力の近似的なものとみなせる)**
- 前提 りづらい (Ott et al., ICML 2018)
- 仮説 高いところしか見ていないので,
  を適切にカバーしていないのではないか?
- 提案 に noise を乗せる
- 具体
  1. テップで)
  2. ormalize, この分布から sampling
  3. ise (word drop, blank, shuffle) を乗せる

例: greedy



Synthetic src の作り方に一工夫

- 一般に, BT は synthetic src 作成に greedy or beam search を行う
  → これらの decoding は最も確率の大きいもの (文, 単語) を出力としている
  (MAP 推定の出力の近似的なものとみなせる)

- 前提: MAP 推定による出力は rich translation になりづらい (Ott et al., ICML 2018)

- 仮説: greedy or beam search は**出力分布の一番高いところしか見ていない**ので,
  生成される **synthetic src は真のデータ分布を適切にカバーしていない**のではないか?
  → synthetic src として regular なものを返すのではないか?

- 提案: 出力分布から sampling する or beam 出力に noise を乗せる

- 具体的にはイカの3種の手法を試した
  1. sampling: 出力分布に基づいて sa
  2. top-k: 出力分布のうち上位 k tok
  3. beam+noise: beam 出力に対して

気持ちとしては分布の一部を見るのではなく,
もっと多様な感じになるようにしている?
🤔: 離散的にそれをやるのは…どうなんだろう…
beam+noise はそうなるのか…?

る

Syn

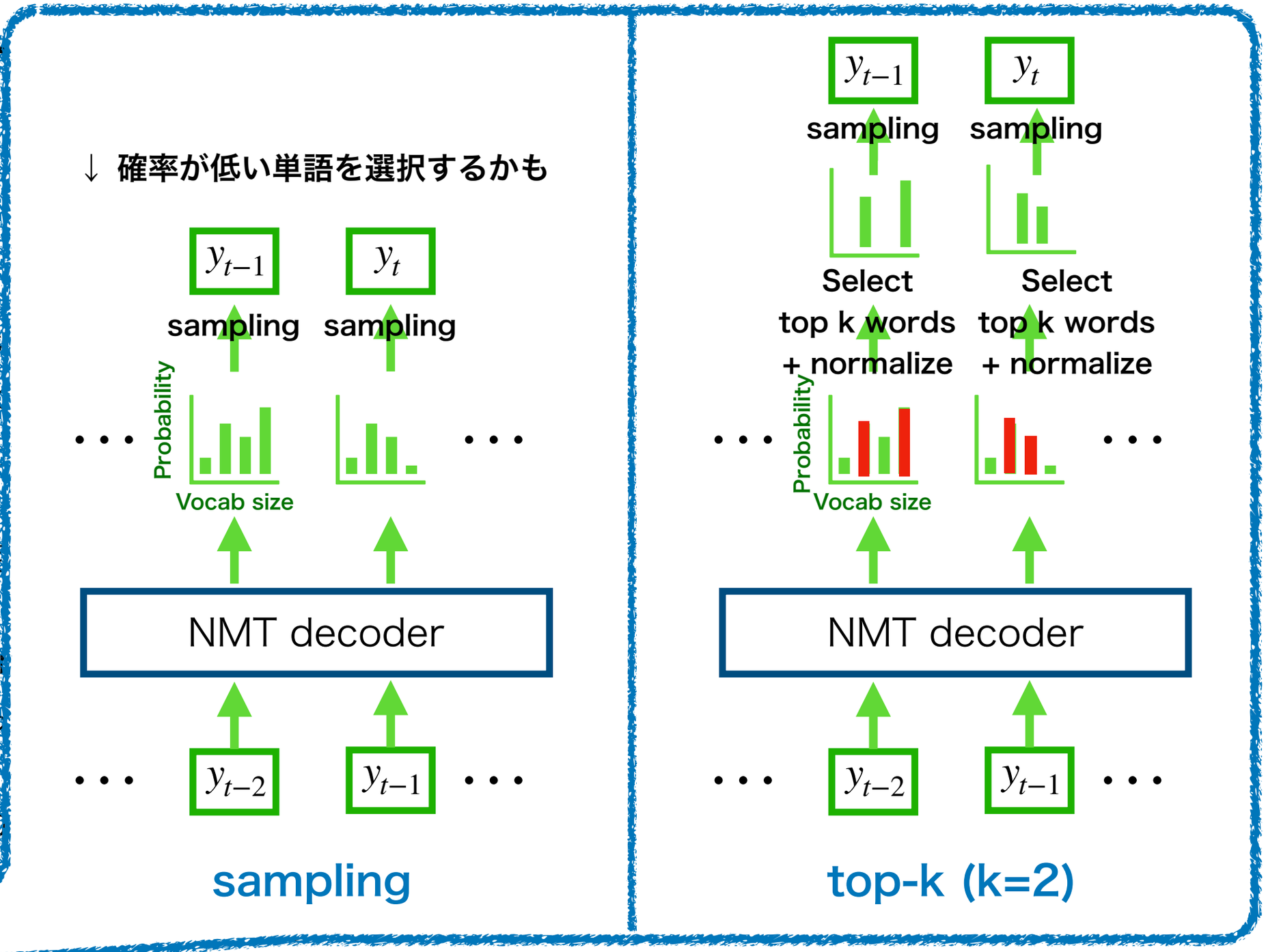
- 一般に, BT は s
  → これらの dec
  (MAP 推定の出

- 前提: MAP 推定

- 仮説: greedy o
  生成される

- 提案: 出力分布か

- 具体的にはイ



1. **sampling**: 出力分布に基づいて sampling (各ステップで)
2. **top-k**: 出力分布のうち上位 k token をとり, renormalize, この分布から sampling
3. beam+noise: beam 出力に対して, 人工的な noise (word drop, blank, shuffle) を乗せる

Synthetic src の作り方に一工夫

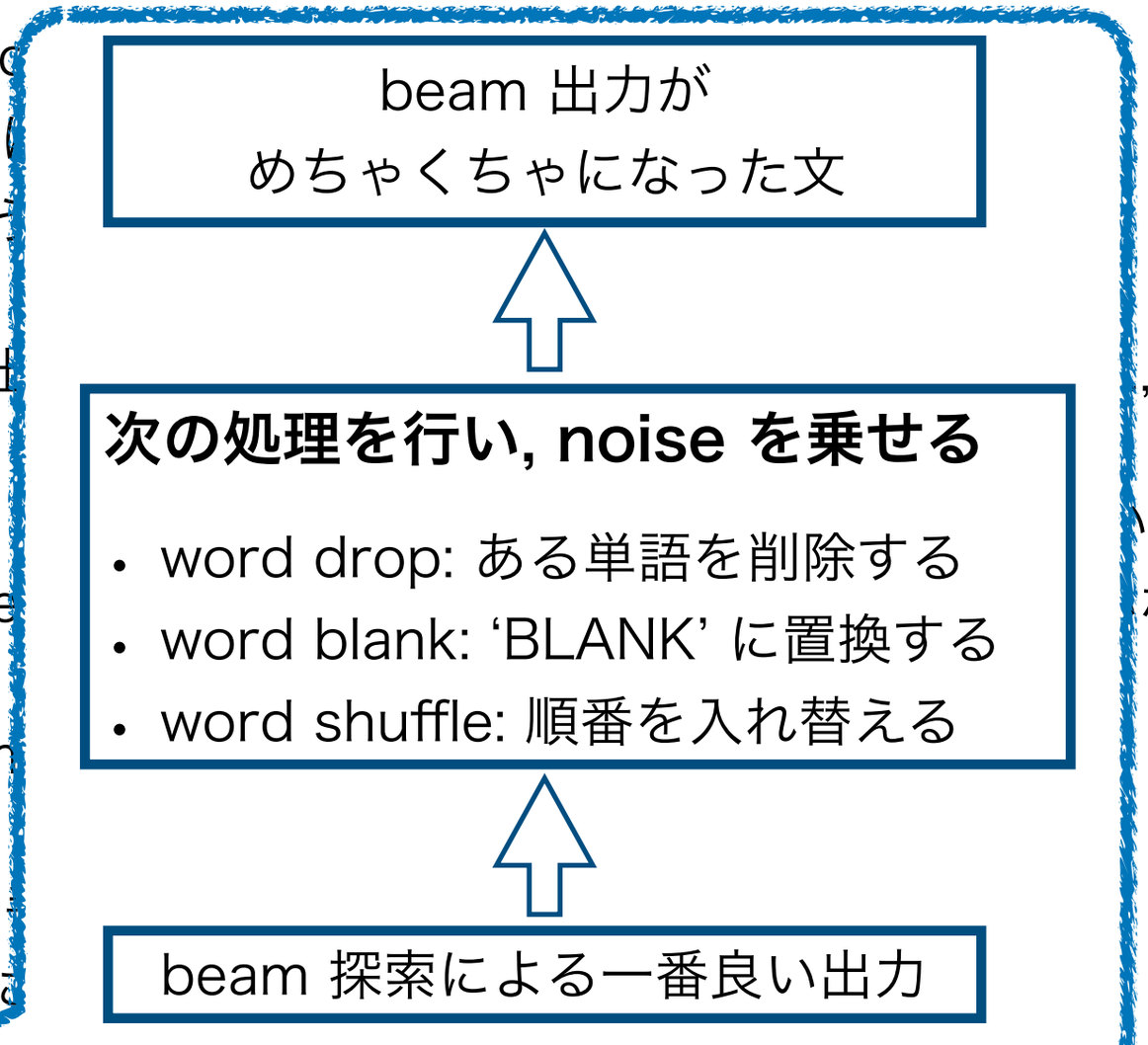
- 一般に, BT は synthetic
  → これらの decoding (
  (MAP 推定の出力のよう

- 前提: MAP 推定による出 , ICML 2018)

- 仮説: greedy or beam ないので,
  生成される synthe ないのではないか?

- 提案: 出力分布から sam

- 具体的にはイカの3種の
  1. sampling: 出力分布に
  2. top-k: 出力分布の から sampling
  3. **beam+noise**: beam 出力に対して, 人工的な noise (word drop, blank, shuffle) を乗せる



Synthetic src

この synthetic src を学習に使用する
→ de-noising Auto Encoder

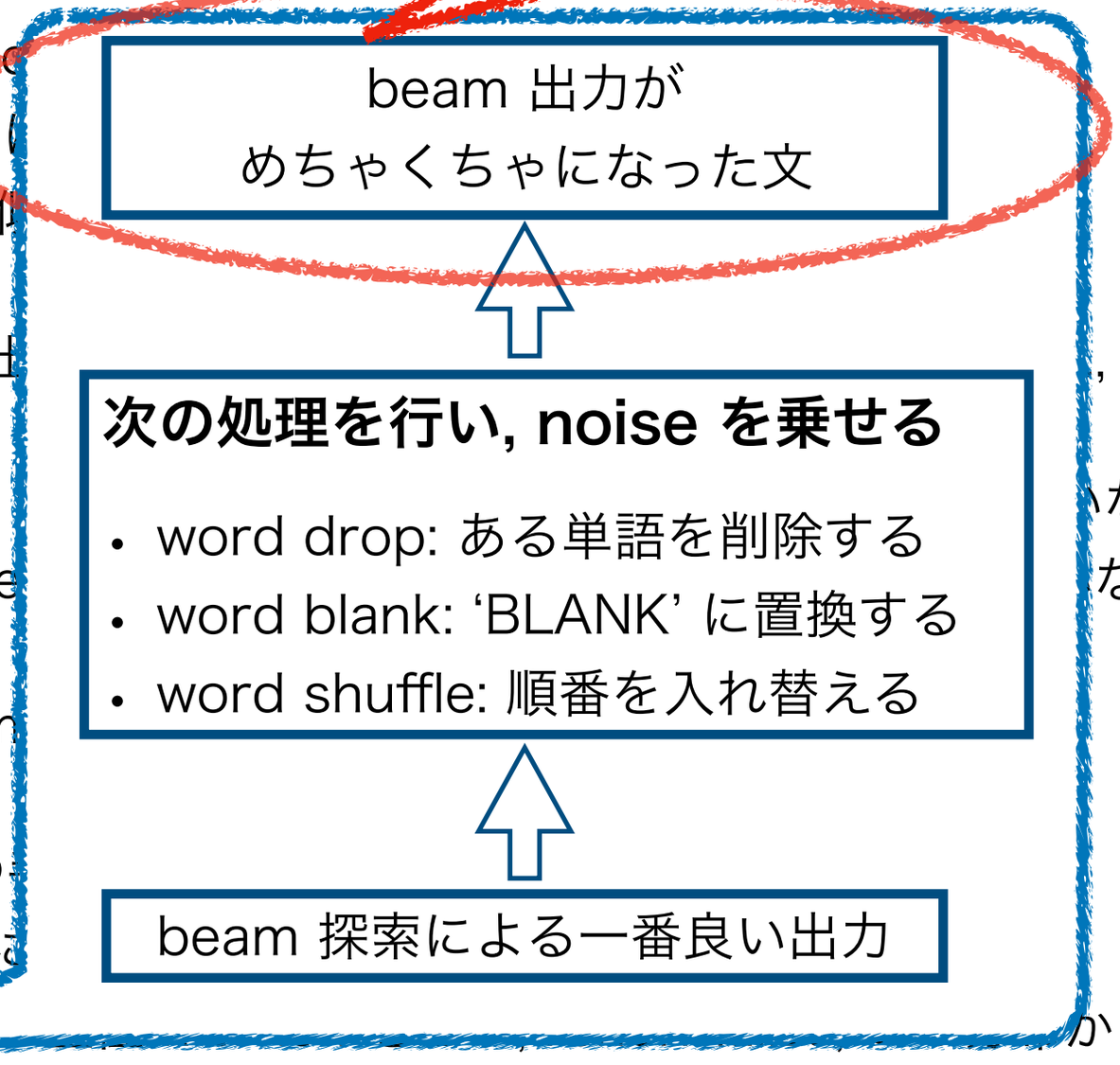
- 一般に, BT は synthetic
  → これらの decoding (
  (MAP 推定の出力の近似

- 前提: MAP 推定による出 , ICML 2018)

- 仮説: greedy or beam ないので,
  生成される synthe ないのではないか?

- 提案: 出力分布から sam

- 具体的にはイカの3種の
  1. sampling: 出力分布に
  2. top-k: 出力分布の から sampling
  3. **beam+noise**: beam 出力に対して, 人工的な noise (word drop, blank, shuffle) を乗せる



Synthetic src の作り方に一工夫

🤔: sample と beam+noise の出力がなんだかがばいことになるのでは?

→ なりました。

(ぶっちゃけ sampling は auto-regressive を考えるともっとえらいことになってもいいと思うが, その辺は語られてない)

source	Diese gegenstzlichen Auffassungen von Fairness liegen nicht nur der politischen Debatte zugrunde.
reference	These competing principles of fairness underlie not only the political debate.
beam	These conflicting interpretations of fairness are not solely based on the political debate.
sample	*Mr President,* these contradictory interpretations of fairness are not based solely on the political debate.
top10	Those conflicting interpretations of fairness are not solely at the heart of the political debate.
beam+noise	conflicting BLANK interpretations BLANK are of not BLANK based on the political debate.

個人的には top-k だと k の幅で多様な出力を出しつつ, そこそこ崩れなくていいのかな, と思っているんですが…

実験設定

- 翻訳方向, data: English-German, WMT'18 ← main
  data size: [parallel] 5.18M sents; [monolingual] 226M sents (max)
  dev: newstest2012
  test: newstest2013-2017
  学習データ (parallel) の 52K sents を held-out set としてる ← 擬似データを入れた時の更新回数を決めてる

- BPE (joint src and tgt): 35K types ← sampling ですごい出力が出てきそう

- 翻訳方向, data: English-French, WMT'14 ← SOTA が言いたい
  data size: [parallel] 35.7M sents; [monolingual] 31M
  dev: newstest2012
  test: newstest2013-2015

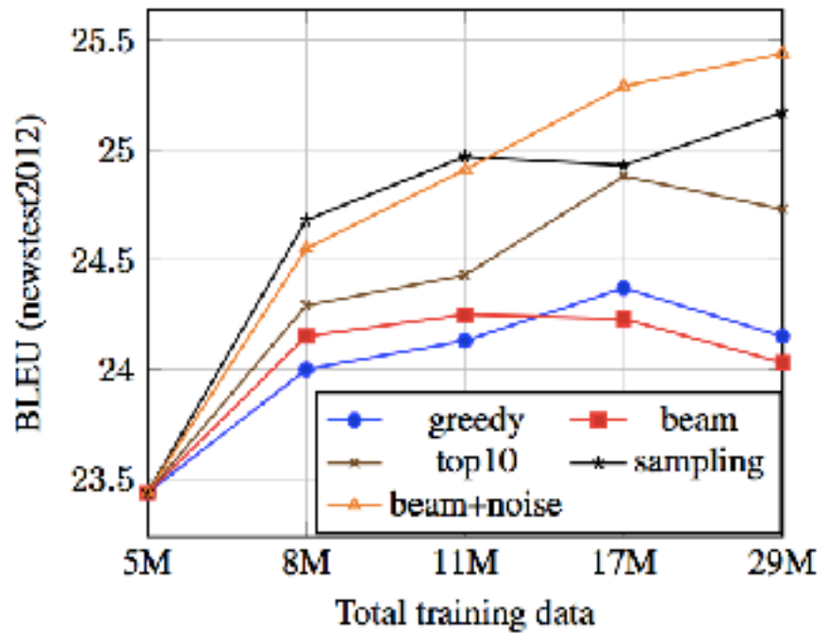
- 評価: tokenized BLEU と detokenized BLEU どちらもやった

- NMT モデル: Big Transformer (fairseq)
  beam size: 5

- Top-k: k= {5,10, 20, 50} でやって, どれも似たような結果になったので **k=10** を報告

Synthetic data の作り方を色々やった結果

5M が bitext only, そこに擬似データを足してる. →



↓ bitext 5.2M sents + synthetic sents 24M

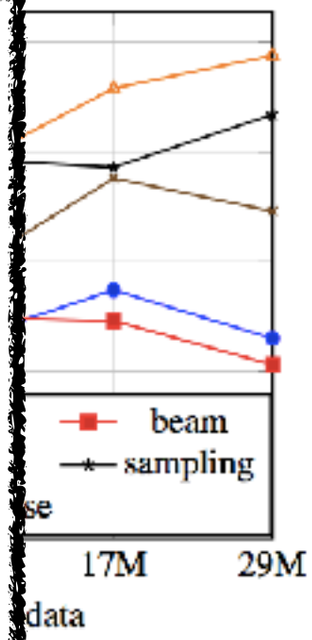
	news2013	news2014	news2015	news2016	news2017	Average
bitext	27.84	30.88	31.82	34.98	29.46	31.00
+ beam	27.82	32.33	32.20	35.43	31.11	31.78
+ greedy	27.67	32.55	32.57	35.74	31.25	31.96
+ top10	28.25	33.94	34.00	36.45	32.08	32.94
+ sampling	28.81	34.46	34.87	37.08	32.35	33.51
+ beam+noise	29.28	33.53	33.79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

Table 1: Tokenized BLEU on various test sets of WMT English-German when adding 24M synthetic sentence pairs obtained by various generation methods to a 5.2M sentence-pair bitext (cf. Figure 1).

Synthetic data の作り方を色々やった結果

sampling と beam+noise がいい感じだった.
top10 はちょっといい.
→ なんでだろう?

noisy src sentence は学習が hard. (denoising AE 並感)
Sampling 出力は argmax 出力より分布をよく表現できている.
→ どちらも **training signal** が beem や greedy より richer なのでは?
→ その辺りを分析する.



	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

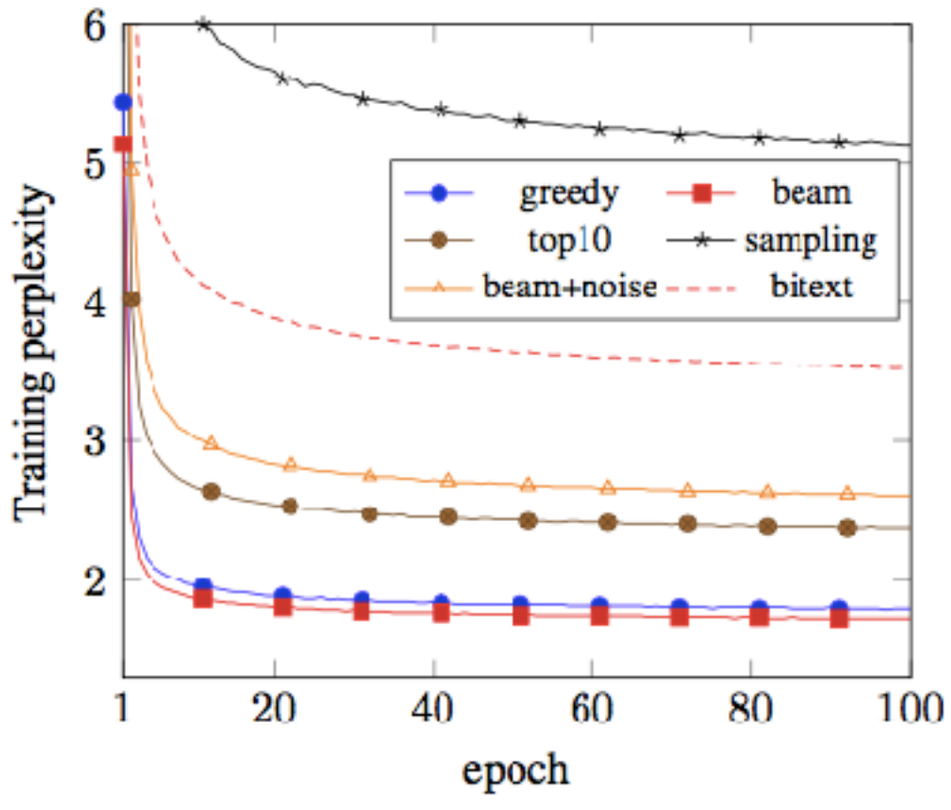
Table 1: Tokenized BLEU on various test sets of WMT English-German when adding 24M synthetic sentence pairs obtained by various generation methods to a 5.2M sentence-pair bitext (cf. Figure 1).

richer training signal なのかを調べて見た with PPL

training loss (ppl) をみてみた (synthetic data: 24M sents)

各 epoch 終了時に training data の 500K 文対に関する ppl
- bitext only は bitext から 500K;
- BT 有りは synthetic data から 500K (tgt 側として同一のものを使用)

→ ppl が即下がる場合は学習がちょろい, ということになる



1. Sampling や beam+noise は ppl が大きい
→ richer training signal, ということらしい
🤔: そもそも何をもって **rich** training signal と言うのか. ppl bigger → richer は言える…?

2. sampling や noise によって多様な src を見る
→ model は reordering や substitution に robust になる (たとえ擬似 noise が現実に存在しなくても)
🤔: 部分的には有りそうだけども…それにしても上がりすぎでは?

richer training signal なのかを調べて見た with 言語モデル

- めっちゃ regular な 学習 data は Language Model で当てやすいのでは
  (つまり多様な感じか regular な感じかを評価する)

- 要するに
  LM を bitext (tgt) で学習,
  分野の同じ bitext で BT を学習,
  この BT によって生成された synthetic src を LM で測る
  low ppl → regular synthetic src

- 実際には, bitext から
  LM 学習の学習に 4.1M sents (src 側) (5-gram; Kneser-Ney)
  BT モデル学習に 640K sents,
  450K に対して BT, 得られた synthetic src に対して LM の ppl を計測

richer training signal なのかを調べて見た with 言語モデル

- めっちゃ regular な 学習 data は Language Model で当てやすいのでは (つまり多様な感じか regular な感じかを評価する)
- 要
LM
分
こ
lov
- bit
LM
BT

- この結果から, sampling や beam+noise の方が rich な出力を出している, と言える
- 単純な beam では 変動性が不足してて noise とかよりも training signal が弱いみたいなことが言えそうですね.

と言うことらしいが…

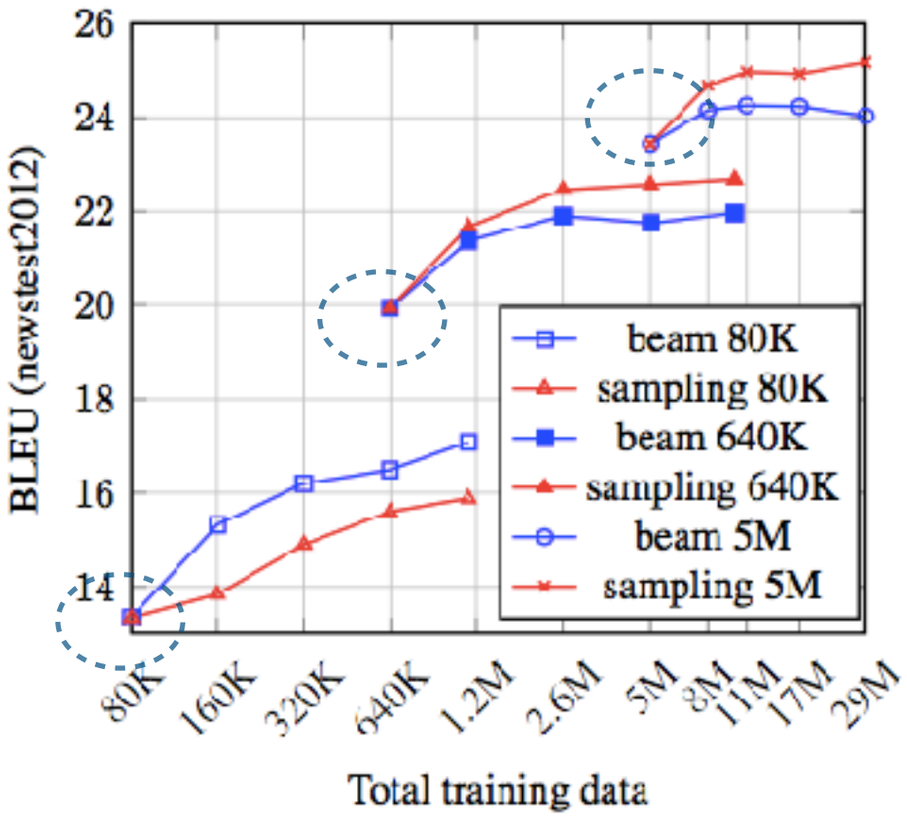
	Perplexity
human data	75.34
beam	72.42
sampling	500.17
top10	87.15
beam+noise	2823.73

450K に対して BT, 得られた synthetic src に対して LM の ppl を計測

Low-resource で効くとめっちゃ嬉しくないですか?

- 今までは bitext として 5M ぐらい使ってきた
  → 減らした時に, この noise を含んだ synthetic src は有効か?

- bitext を 80K 使う設定, 640K 使う設定でやって見た
  BT (De-En) は 80K で 13.5 BLEU, 640K で 24.3 BLEU, 5M で 28.3 BLEU



← 点線の丸で囲んだ点が bitext only
そこから synthetic src を足してる形

結果: low-resource だと sampling で増やすよりは beam で増やした方が良さそう
ていうのも, きちんとしたコーパスが小さいとそもそも noise に対して脆い感じになるから.

Synthetic data の domain 事情

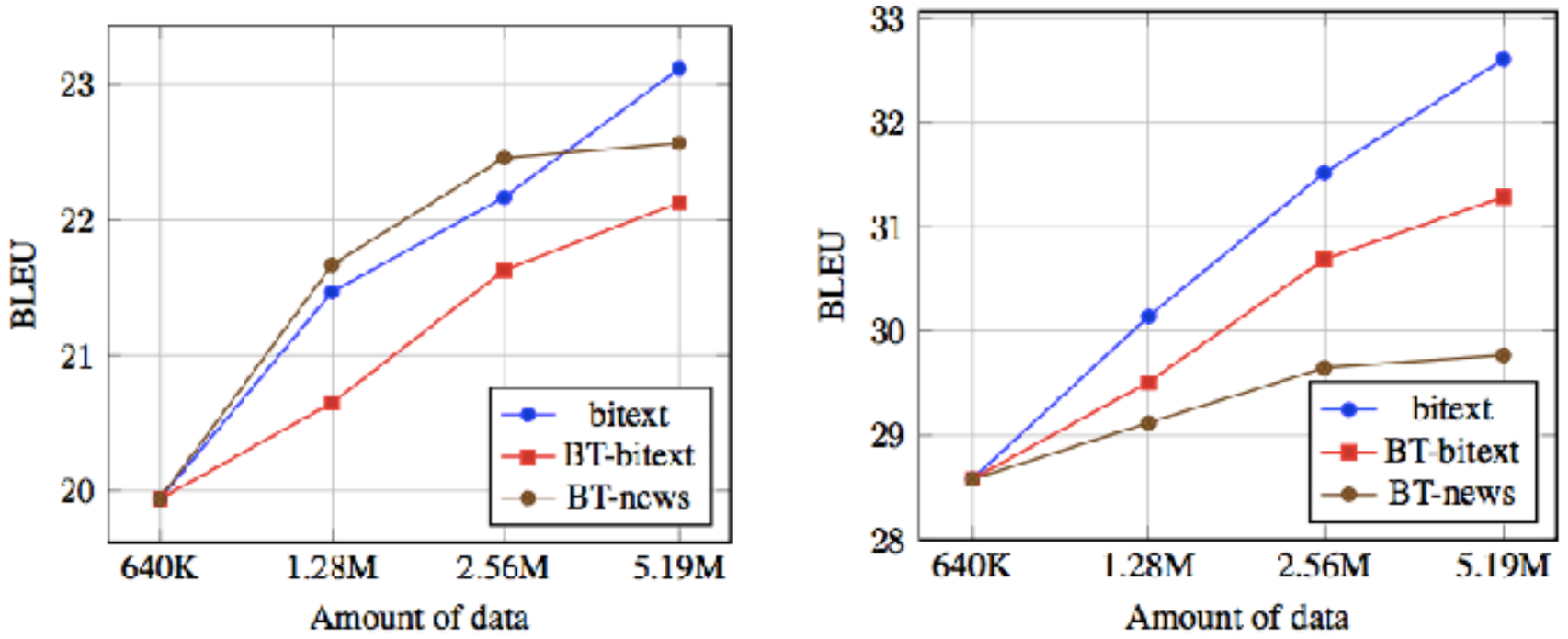
- 本物 vs 偽物, Domain とかの影響調査する

BT は sampling を行う

- Bitext から 640K sents pair を使って BT を学習する
  さらにこの 640K sents pair と
  1. 残りの parallel data を使用して順方向を学習 (normal bitext)
  2. 残りの parallel data の tgt に対して BT による推論を行い, その synthetic data を使用して順方向を学習 (BT-bitext)
  3. Newscrawl data に対して BT による推論を行い, その synthetic data を使用して順方向を学習 (BT-news)

- 1 と 2 の比較で 本物 vs 偽物 を,
  2 と 3 の比較で, domain に関する影響調査を行う.

- 評価 set: newstest2012 (newswire) と WMT の held-out set (mix)



(a) newstest2012 (b) valid-mixed

Figure 4: Accuracy on (a) newstest2012 and (b) a mixed domain valid set when growing a 640K bitext corpus with (i) real parallel data (bitext), (ii) a back-translated version of the target side of the bitext (BT-bitext), (iii) or back-translated newscrawl data (BT-news).

1. 本物 vs 偽物
   a → 本物 (bitext) が 2.6 BLEU up; 偽物 (BT-bitext) が 2.2 BLEU up
   → 本物に対して83%まで達成!
2. Domain 調査
   a, b の BT-news, BT-bitext の改善度合いをみてみる → 揃っている方が良い
   a から十分量の同一 domain monolingual があれば, もともと mix で学習 (bitext) した場合よりもいい感じ (domain adaptation 的なことはまあできる)

bitext の方を upsampling して見た

- Synthetic data と data size を揃えるような気持ち
- 具体的にはもともと bitext 5M だったものを n倍する感じ
  例: n = 4
  → bitext 20M, synthetic 24M
- Fig 5 が nを1, 2, 4, 8 にした時の結果
  Synthetic data は 24M 固定
- Greedy や beam は nをあげると良いが, noise 系はそうでもない

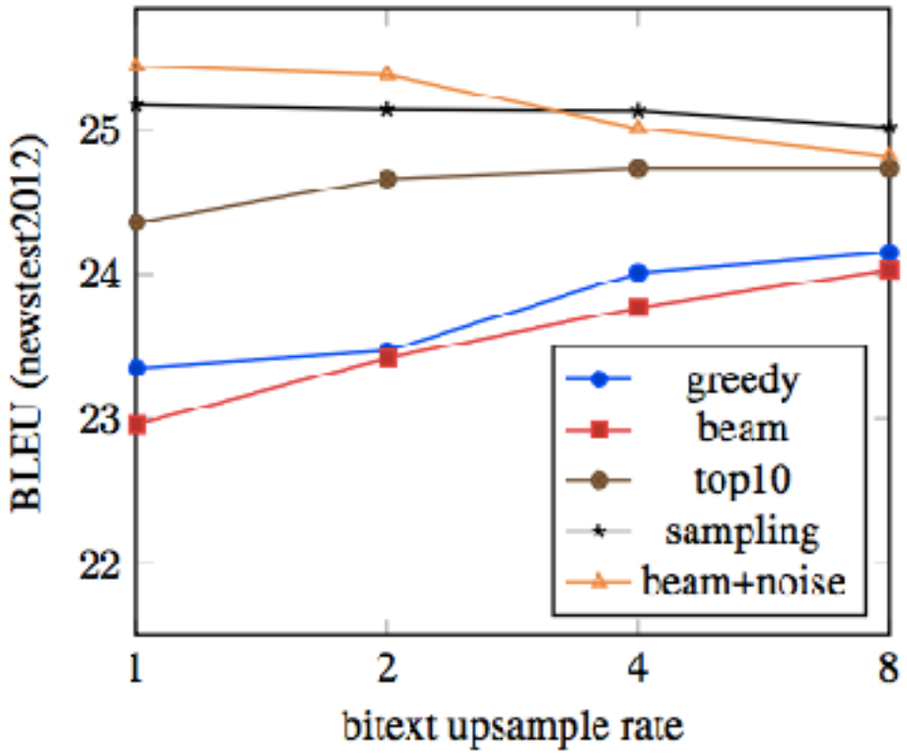


Figure 5: Accuracy when changing the rate at which the bitext is upsampled during training. Rates larger than one mean that the bitext is observed more often than actually present in the combined bitext and synthetic training corpus.

SOTA とった話

- monolingual data として 226M sents 突っ込んだ
  → とりました

	En–De	En–Fr
a. Gehring et al. (2017)	25.2	40.5
b. Vaswani et al. (2017)	28.4	41.0
c. Ahmed et al. (2017)	28.9	41.4
d. Shaw et al. (2018)	29.2	41.5
DeepL	33.3	**45.9**
Our result	**35.0**	45.6
detok. sacreBLEU[3]	*33.8*	*43.8*

Table 6: BLEU on newstest2014 for WMT English-German (En–De) and English-French (En–Fr). The first four results use only WMT bitext (WMT'14, except for b, c, d in En–De which train on WMT'16). DeepL uses proprietary high-quality bitext and our result relies on back-translation with 226M newscrawl sentences for En–De and 31M for En–Fr. We also show detokenized BLEU (SacreBLEU).

まとめ

- Back translation による出力を noisy な感じにすることで性能が上がった.
  → 単純な従来の出力と比べて, richer training signal を出してるからでしょう. (ただし low-resource は除く)

- SOTA とった

- 🤔: 色々雑だけどもとりあえず大量データがあるなら離散的 noise (src 側のみ) は割と有効?
  最初は n-best 出力, normalize, sample sentence みたいにすると思ってた勢

参考文献

- Improving Neural Machine Translation Models with Monolingual Data. Sennrich et al., ACL2016

- Phrase-Based & Neural Unsupervised Machine Translation. Lample et al., EMNLP2018

- Analyzing Uncertainty in Neural Machine Translation. Ott et al., ICML 2018

- GEC で似た話
  Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. Xie et al., NAACL 2018