ACL-IJCNLP 2021 Papers Review

김백수 조주현

TABLE OF CONTENTS

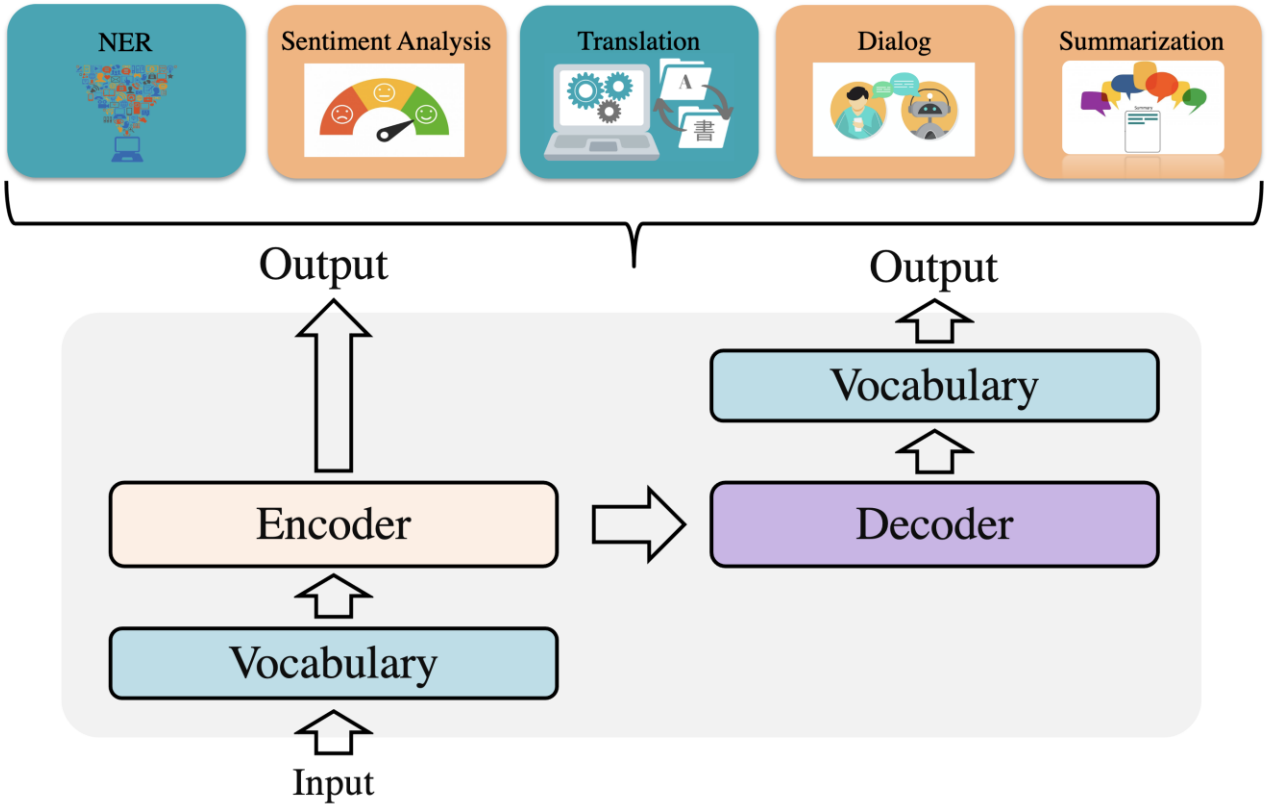
Vocabulary Learning via Optimal Transport for Neural Machine Translation

Vocabulary Learning
via Optimal Transport
for Neural Machine Translation

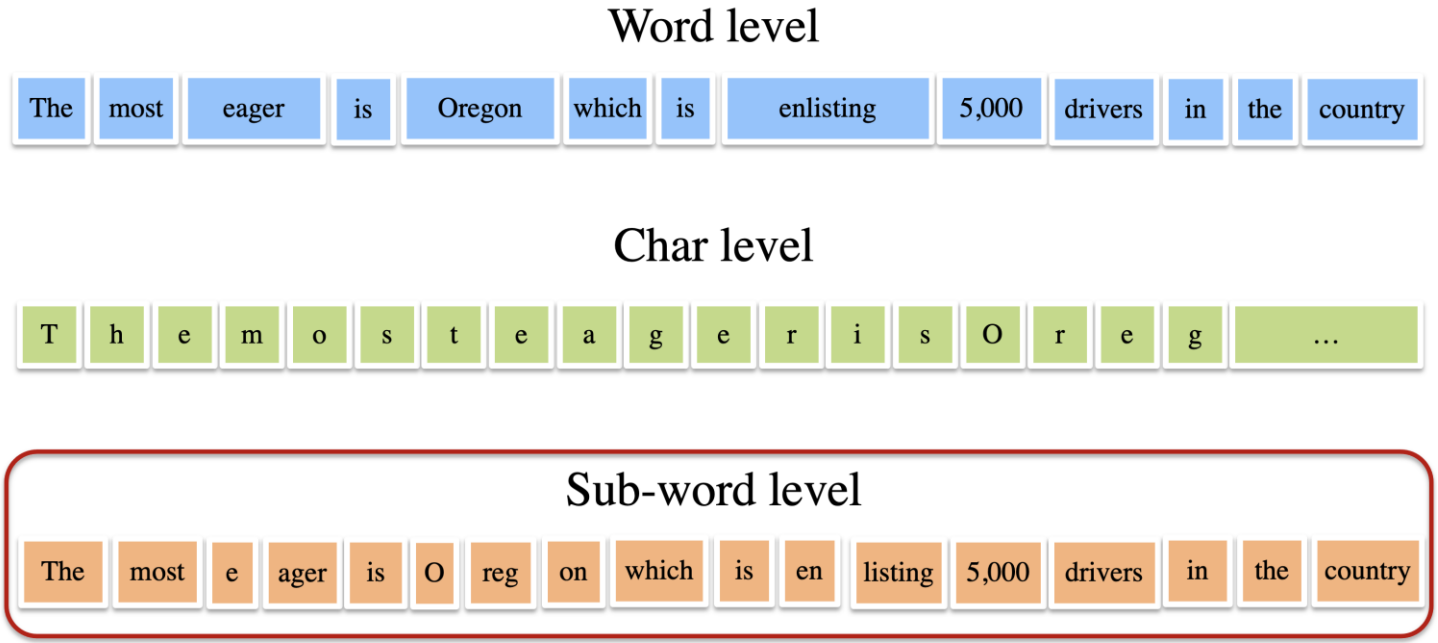
Vocabulary Learning via Optimal Transport for Neural Machine Translation

Jingjing Xu[1], **Hao Zhou**[1], **Chun Gan**[1,2†], **Zaixiang Zheng**[1,3†], **Lei Li**[1]
[1]ByteDance AI Lab
[2]Math Department, University of Wisconsin–Madison
[3]Nanjing University

What to solve



What to solve



Sub-word vocabulary is the dominant choice

Why?

Sub-word vocabularies can be regarded as a trade-off between character-level vocabularies and word-level vocabularies. Compared to word-level vocabularies, it can decrease the sparsity of tokens and increase the shared features between similar words, which probably have similar semantic meanings, like “happy” and “happier”. Compared to character-level vocabularies, it has shorter sentence lengths without rare words.

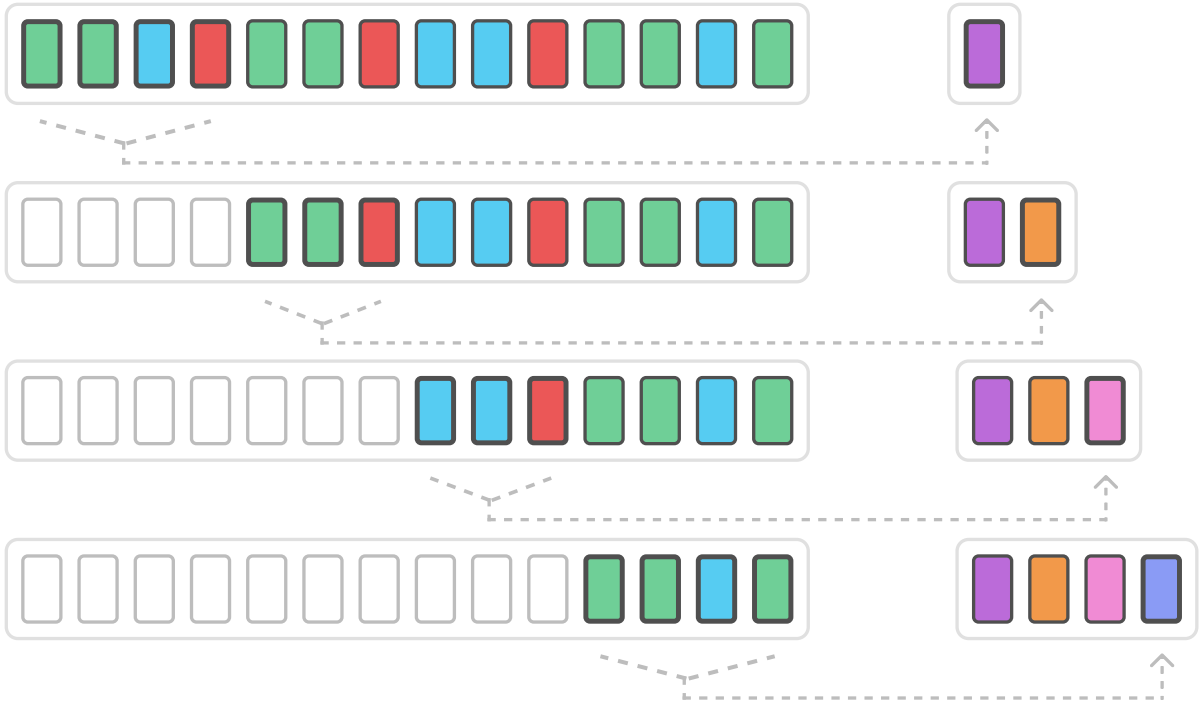
Previous Solutions
BPE

Byte-Pair Encoding

Frequency – Based Encoding

1) Starts from every characters as a vocabulary(token)set
2) Calculate every pair’s frequency
3) Merge most frequent pair as a single token
4) Repeat 2) and 3) for n iteration

Previous Solutions
BPE

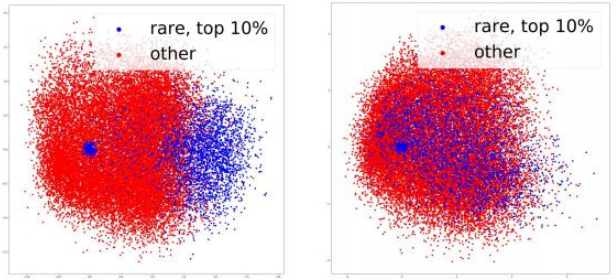


Previous Solutions
BPE-dropout

Byte-Pair Encoding + Dropout on pair list building

Have a strength on

1) *Rare Words*
2) Typo



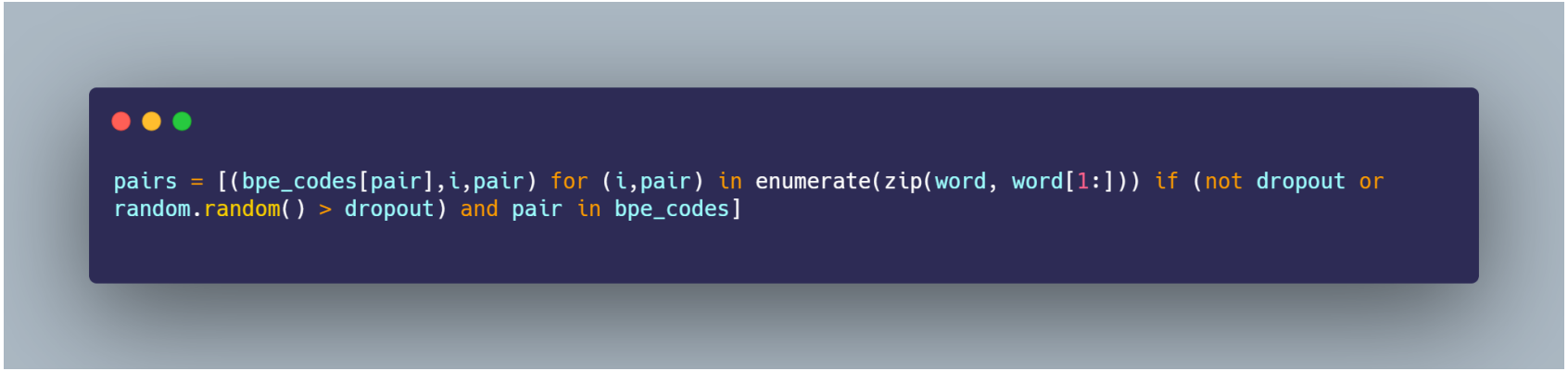
(a) BPE (b) *BPE-dropout*

Figure 7: Visualization of source embeddings. Models trained on WMT14 En-Fr (4m).

source	BPE	*BPE-dropout*	diff
En-De			
original	27.41	**28.01**	+0.6
misspelled	24.45	**26.03**	+1.58
De-En			
original	32.69	**34.19**	+1.5
misspelled	29.71	**32.03**	+2.32
En-Fr (4m)			
original	33.38	**33.85**	+0.47
misspelled	30.30	**32.13**	+1.83
En-Fr (16m)			
original	34.37	**34.82**	+0.45
misspelled	31.23	**32.94**	+1.71

Table 5: BLEU scores for models trained on WMT14 dataset evaluated given the original and misspelled source. For En-Fr trained on 16m sentence pairs, *BPE-dropout* was used only on the source side (Section 5.2).

Previous Solutions
BPE-dropout



Previous Solutions
Sentencepiece

SentencePiece is a re-implementation of **sub-word units**, an effective way to alleviate the open vocabulary problems in neural machine translation. SentencePiece supports two segmentation algorithms, **byte-pair-encoding (BPE)** [Sennrich et al.] and **unigram language model** [Kudo.]. Here are the high level differences from other implementations.

Neural Machine Translation models typically operate with a fixed vocabulary. Unlike most unsupervised word segmentation algorithms, which assume an infinite vocabulary, SentencePiece trains the segmentation model such that the final vocabulary size is fixed, e.g., 8k, 16k, or 32k.

Note that SentencePiece specifies the final vocabulary size for training, which is different from subword-nmt that uses the number of merge operations. The number of merge operations is a BPE-specific parameter and not applicable to other segmentation algorithms, including unigram, word and character.

Main Problem

Previous approach cares Frequency only.

How about size?

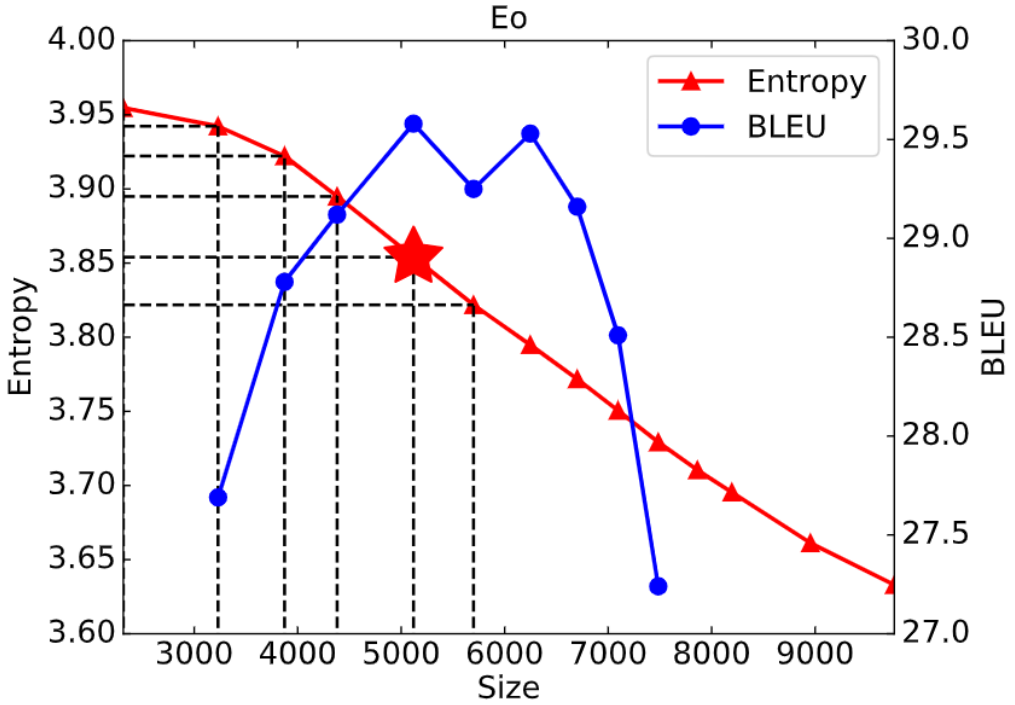
Despite promising results, most existing subword approaches only consider frequency (or entropy) while the effects of vocabulary size is neglected. Thus, trial training is required to find the optimal size, which brings high computation costs. In this work, we aim to figure out how to evaluate vocabularies and whether one can find the optimal vocabulary without trial training. In the next, I will introduce our work following these two questions.

Metrics: Size & Entropy

Size is an essential factor. From the perspective of size, a vocabulary with smaller size is a better choice. Smaller size usually means less rare tokens and less parameters.

Entropy is also an important factor. Currently, sub-word approaches like Byte-Pair Encoding (BPE) are widely used in the community. In information theory, BPE are simple forms of data compression. The target is to reduce the entropy (or bits-per-char) of a corpus, which uses fewer bits to represent the corpus than the character-level segmentation.

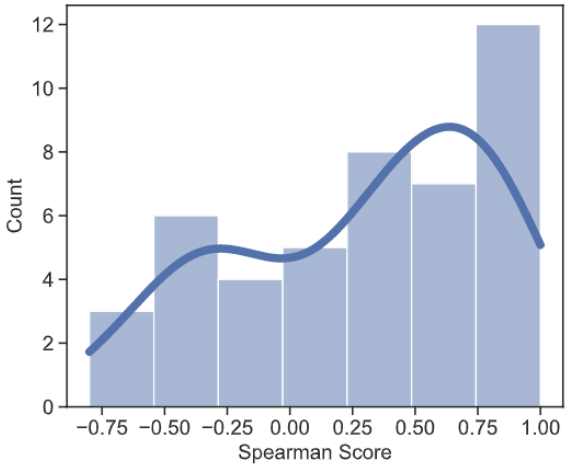
Metrics: Size & Entropy



Key concept1 . Marginal utility

$$\mathcal{M}_{v(k+m)} = \frac{-(\mathcal{H}_{k+m} - \mathcal{H}_{v(k)})}{m}$$

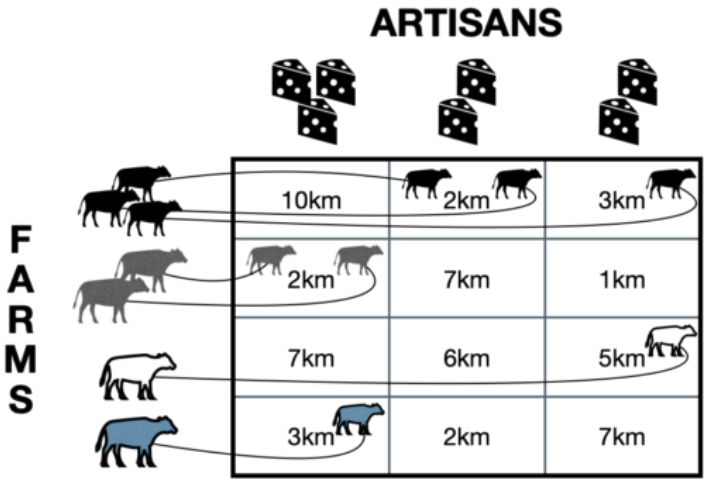
Results. *To verify the effectiveness of MUV as the vocabulary measurement, we conduct experiments on 45 language pairs from TED and calculate the Spearman correlation score between MUV and BLEU scores. We adopt the same and widely-used settings to avoid the effects of other attributes on BLEU scores, such as model hyper-parameters and training hyper-parameters.*



Key concept2 – optimal transport

*The goal then would be to find a function **T**, called the **Monge map**, that assigns each farm to an artisan in an optimal way by taking into account both the distances between them and their respective demands.*

In fact, the true power of OT is hidden in its ability to act as a mechanism of transforming one (continuous) probability distribution into another one with the lowest possible effort.



Key concept2 – optimal transport

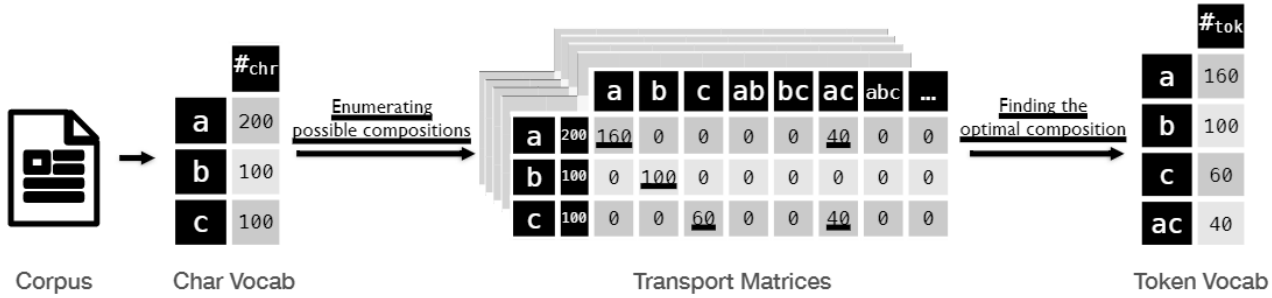


Figure 3: An illustration of vocabulary construction from a transport view. Each transport matrix represents a vocabulary. The transport matrix decides how many chars are transported to token candidates. The tokens with zero chars will not be added into the vocabulary.

Details

$$\underset{v(t-1)\in\mathbb{V}_{S[t-1]},v(t)\in\mathbb{V}_{S[t]}}{\operatorname{argmax}} -\frac{1}{k}(\mathcal{H}_{v(t)} - \mathcal{H}_{v(t-1)})$$

....(some manipulation)

The Upper bound is

$$\underset{v(t)\in\mathbb{V}_{S[t]}}{\operatorname{argmax}}\ \mathcal{H}_{v(t)} - \underset{v(t-1)\in\mathbb{V}_{S[t-1]}}{\operatorname{argmax}}\ \mathcal{H}_{v(t-1)}$$

the only thing we need to do is to find a vocabulary from $\mathbb{V}_{S[t]}$ with the maximum entropy score.

Details

1) searching for the optimal vocabulary with the highest entropy at each timestep t;
2) enumerating all timesteps and outputing the vocabulary corresponding to the time step satisfying Eq

$$\underset{v(t)\in\mathbb{V}_{S[t]}}{\operatorname{argmax}} -\frac{1}{l_{v(t)}}\sum_{j\in v(t)} P(j)logP(j)$$

$$\min_{v(t)\in\mathbb{V}_{S[t]}} \frac{1}{l_v}\sum_{j\in v(t)} P(j)logP(j)$$

$$s.t.P(j)=\frac{Token(j)}{\sum_{j\in v} Token(j)},\quad l_v=\frac{\sum_{j\in v} len(j)}{|v|}$$

$\sum_{j\in\mathbb{T}} P(j)logP(j) = \sum_{j\in\mathbb{C}}\sum_{j\in v} P(i,j)logP(i,j) + \sum_{j\in\mathbb{C}}\sum_{j\in v} P(i,j)(-logP(i|j))$
P : Transportation Matrix / Blue: Objective Function / Red: Distance Matrix

Details

Transport Matrix P

P_a	$P_{a,ab}$	$P_{a,ab}$	$P_{a,a}$
P_b	$P_{b,ab}$	$P_{b,bc}$	$P_{b,a}$
P_c	$P_{c,ab}$	$P_{c,bc}$	$P_{c,a}$
	P_{ab}	P_{bc}	P_a

Distance Matrix D

a	1/2	∞	1/1
b	1/2	1/2	∞
c	∞	1/2	∞
	ab	bc	a

Constraints

$$\forall j \in \{a,b,c\}, \sum_{i \in \{ab,bc,a\}} P_{i,j} = P_j$$

$$\forall i \in \{ab,bc,a\}, \sum_{j \in \{a,b,c\}} P_{i,j} - P_i \leq \epsilon$$

Problem

$$\min_{\text{all } P} C(P)$$

Cost Function

$$C(P) = -H(P) + \sum_{\substack{j \in \{a,b,c\}, \\ i \in \{ab,bc,a\}}} P_{i,j} D_{i,j}$$

Details

Algorithm 1: VOLT

Input: A sequence of token candidates $\mathbb{L}$ ranked by frequencies, an incremental integer sequence $\boldsymbol{S}$ where the last item of $\boldsymbol{S}$ is less than $|\mathbb{L}|$, a character sequence $\mathbb{C}$, a training corpus D_c

Parameters: u $\in \mathbb{R}_+^{|\mathbb{C}|}$, v $\in \mathbb{R}_+^{|\mathbb{T}|}$

vocabularies = []

for *item in* $\boldsymbol{S}$ **do**

- `// Begin of Sinkhorn algorithm`
- Initialize u = ones() and v = ones()
- $\mathbb{T} = \mathbb{L}[: item]$
- Calculate token frequencies $P(\mathbb{T})$ based on D_c
- Calculate char frequencies $P(\mathbb{C})$ based on D_c
- Calculate $\boldsymbol{D}$
- **while** *not converge* **do**
  - u = $P(\mathbb{T})/\boldsymbol{D}$v
  - v = $P(\mathbb{C})/\boldsymbol{D}^T$u
- optimal_matrix = u.reshape(-1, 1) * $\boldsymbol{D}$ * v.reshape(1, -1)
- `// End of Sinkhorn algorithm`
- entropy, vocab = get_vocab(optimal_matrix)
- vocabularies.append(entropy,vocab)

Output $\boldsymbol{v}^*$ from vocabularies satisfying Eq. 3

Experiment

Table 1: Comparison between vocabularies search by VOLT and widely-used BPE vocabularies. VOLT achieves higher BLEU scores with large size reduction. Here the vocabulary size is adopted from the X-En setting.

Bilingual	WMT-14	TED											
En-X	De	Es	PTbr	Fr	Ru	He	Ar	It	Nl	Ro	Tr	De	Vi
BPE-30K	29.31	39.57	39.95	40.11	19.79	26.52	16.27	34.61	32.48	27.65	15.15	29.37	28.20
VOLT	**29.80**	**39.97**	**40.47**	**40.42**	**20.36**	**27.98**	**16.96**	**34.64**	**32.59**	**28.08**	**16.17**	**29.98**	**28.52**
X-En	De	Es	PTbr	Fr	Ru	He	Ar	It	Nl	Ro	Tr	De	Vi
BPE-30K	**32.60**	**42.59**	45.12	**40.72**	24.95	37.49	31.45	38.79	37.01	35.60	25.70	36.36	27.48
VOLT	32.30	42.34	**45.93**	**40.72**	**25.33**	**38.70**	**32.97**	**39.09**	**37.31**	**36.53**	**26.75**	**36.68**	**27.39**
Vocab Size (K)	De	Es	PTbr	Fr	Ru	He	Ar	It	Nl	Ro	Tr	De	Vi
BPE-30K	33.6	29.9	29.8	29.8	30.1	30.0	30.3	33.5	29.8	29.8	29.9	30.0	29.9
VOLT	**11.6**	**5.3**	**5.2**	**9.2**	**3.3**	**7.3**	**9.4**	**3.2**	**2.4**	**3.2**	**7.2**	**8.2**	**8.4**

Table 2: Comparison between vocabularies search by VOLT and BPE-1K, recommended by Ding et al. (2019) for low-resource datasets. Here we take TED X-En bilingual translation as an example. This table demonstrates that vocabularies searched by VOLT are on par with heuristically-searched vocabularies in terms of BLEU scores.

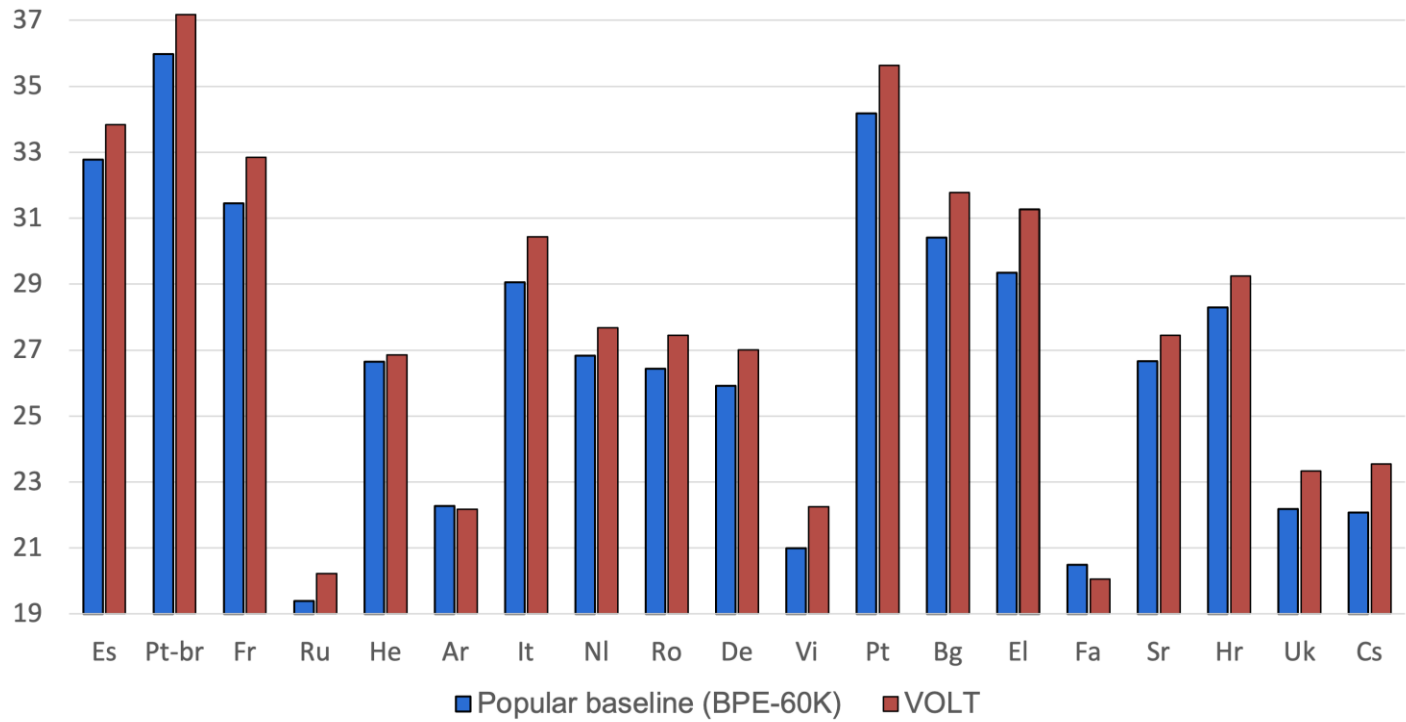
X-En	Es	PTbr	Fr	Ru	He	Ar	It	Nl	Ro	Tr	De	Vi	Avg
BPE-1K	**42.36**	45.58	**40.90**	24.94	38.62	32.23	38.75	**37.44**	35.74	25.94	**37.00**	27.28	35.65
VOLT	42.34	**45.93**	40.72	**25.33**	**38.70**	**32.97**	**39.09**	37.31	**36.53**	**26.75**	36.68	**27.39**	**35.81**
Vocab Size (K)	Es	PTbr	Fr	Ru	He	Ar	Ko	It	Nl	Ro	Tr	De	Avg
BPE-1K	**1.4**	**1.3**	**1.3**	**1.4**	**1.3**	**1.5**	**4.7**	**1.2**	**1.2**	**1.2**	**1.2**	**1.2**	**1.6**
VOLT	5.3	5.2	9.2	3.3	7.3	9.4	3.2	2.4	3.2	7.2	8.2	8.4	6.0

Experiment

Table 3: Comparison between VOLT and widely-used BPE vocabularies on multilingual translation. VOLT achieves higher BLEU scores on most pairs.

X-En	Es	Pt-br	Fr	Ru	He	Ar	Ko	Zh-cn	It	Ja	Zh-tw	Nl	Ro
BPE-60K	32.77	35.97	31.45	19.39	26.65	**22.28**	14.13	15.80	29.06	10.31	15.03	26.83	26.44
VOLT	**33.84**	**37.18**	**32.85**	**20.23**	**26.85**	22.17	**14.36**	**16.59**	**30.44**	**10.75**	**15.73**	**27.68**	**27.45**
X-En	Tr	De	Vi	Pl	Pt	Bg	El	Fa	Sr	Hu	Hr	Uk	Cs
BPE-60K	16.74	25.92	21.00	18.06	34.17	30.41	29.35	20.49	26.66	17.97	28.30	22.18	22.08
VOLT	**17.55**	**27.01**	**22.25**	**18.93**	**35.64**	**31.77**	**31.27**	**20.05**	**27.45**	**19.00**	**29.25**	**23.34**	**23.54**
X-En	Id	Th	Sv	Sk	Sq	Lt	Da	My	Sl	Mk	Fr-ca	Fi	Hy
BPE-60K	24.58	17.92	30.43	24.68	28.50	19.17	34.65	13.54	20.59	28.23	27.20	15.13	17.68
VOLT	**25.87**	**18.89**	**31.47**	**25.69**	**29.09**	**19.85**	**36.04**	**13.65**	**21.36**	**28.54**	**28.35**	**15.98**	**18.44**
X-En	Hi	Nb	Ka	Mn	Et	Ku	Gl	Mr	Zh	Ur	Eo	Ms	Az
BPE-60K	**18.57**	**35.96**	**16.47**	**7.96**	15.91	**13.39**	26.75	**8.94**	**13.35**	**14.21**	**21.66**	**19.82**	**9.67**
VOLT	18.54	35.88	15.97	**7.96**	**16.03**	13.20	**26.94**	8.40	12.67	13.89	21.43	19.06	9.09

Experiment


37
35
33
31
29
27
25
23
21
19
Es
Pt-br
Fr
Ru
He
Ar
It
Nl
Ro
De
Vi
Pt
Bg
El
Fa
Sr
Hr
Uk
Cs
Popular baseline (BPE-60K)
VOLT

Experiment

Table 4: Results of VOLT, MUV-Search and BPE-Search. MUV-Search does not require full training and saves a lot of costs. Among them, VOLT is the most efficient solution. MUV-Search and VOLT require additional costs for downstream evaluation, which takes around 32 GPU hours. "GH" and "CH" represent GPU hours and CPU hours, respectively.

En-De	BLEU	Size	Cost
BPE-Search	**29.9**	12.6K	384 GH
MUV-Search	29.7	**9.70K**	5.4 CH + 30 GH
VOLT	29.8	11.6K	**0.5 CH + 30 GH**

Table 5: Comparison between VOLT and strong baselines. VOLT achieves almost the best performance with a much smaller vocabulary.

En-De	BLEU	Parameters
(Vaswani et al., 2017)	28.4	210M
(Shaw et al., 2018)	29.2	213M
(Ott et al., 2018)	29.3	210M
(So et al., 2019)	29.8	218M
(Liu et al., 2020)	**30.1**	256M
SentencePiece	28.7	210M
WordPiece	29.0	210M
VOLT	29.8	**188M**

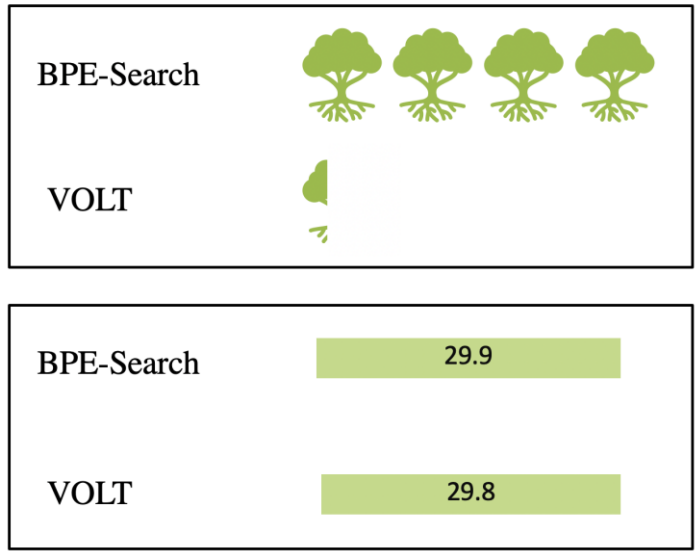
Experiment

Table 6: Vocabularies searched by VOLT are better than widely-used vocabularies on various architectures. Here "better" means competitive results but much smaller sizes.

En-De	Approach	BLEU	Size
Transformer-big	BPE-30K	29.3	33.6K
	VOLT	**29.8**	**11.6K**
Convolutional Seq2Seq	BPE-30K	**26.4**	33.6K
	VOLT	26.3	**11.6K**

Experiment

One advantage of VOLT lies in its low resource consumption. We first compare VOLT with BPE-Search, a method to select the best one from a BPE-generated vocabulary set based on their BLEU scores. In BPE-Search, we first define a vocabulary set including BPE-1K, BPE-2K, BPE-3K, BPE-4K, BPE-5K, BPE-6K, BPE-7K, BPE-8K, BPE-9K, BPE-10K, BPE-20K, BPE-30K. Then, we run full experiments to select the best vocabulary. The cost of BPE-Search is the sum of the training time on all vocabularies. VOLT is a lightweight solution that can find a competitive vocabulary with much less computation requirements.



references

https://jingjing-nlp.github.io/volt-blog/

https://towardsdatascience.com/optimal-transport-a-hidden-gem-that-empowers-todays-machine-learning-2609bbf67e59

https://github.com/Jingjing-NLP/VOLT

UnNatural Language Inference

UnNatural Language Inference

Koustuv Sinha[1,2,3], Prasanna Parthasarathi[1,2], Joelle Pineau[1,2,3] and Adina Williams[3]
[1] School of Computer Science, McGill University, Canada
[2] Montreal Institute of Learning Algorithms (Mila), Canada
[3] Facebook AI Research (FAIR)
{koustuv.sinha, prasanna.parthasarathi, jpineau, adinawilliams}
@{mail.mcgill.ca, mail.mcgill.ca, cs.mcgill.ca, fb.com}

“Pretrained LMs know sy

- Wu et al. (2020) recover syntactic trees from BERT considering attention patterns
- Tenney et al. (2019) conclude that BERT ‘recreates the classical NLP pipeline:’ POS tagging, parsing, NER, semantic roles, coreference...
- Many papers claim **LMs “know syntax”** on the basis of probes and diagnostic datasets

(Goldberg, 2019; Hewitt and Manning, 2019; Jawahar et al., 2019; Wu et al., 2020; Warstadt et al 2019a,b; Warstadt and Bowman 2020; Linzen and Baroni 2021...)

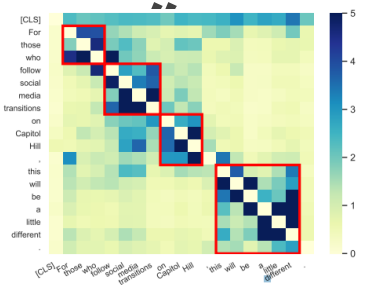


Figure 1: Heatmap of the impact matrix for the sentence “For those who follow social media transitions on Capitol Hill, this will be a little different.”

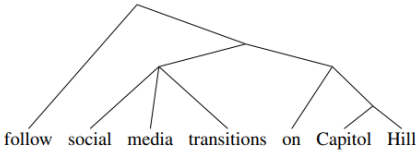


Figure 2: Part of the constituency tree.

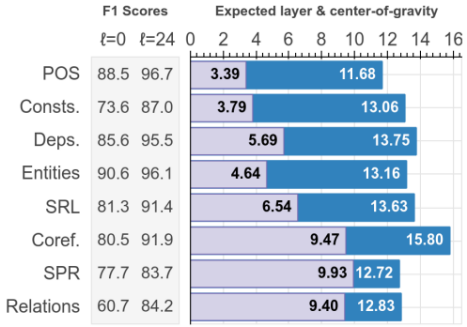


Figure 1: Summary statistics on BERT-large. Columns on left show F1 dev-set scores for the baseline ($P_\tau^{(0)}$) and full-model ($P_\tau^{(L)}$) probes. Dark (blue) are the mixing weight center of gravity (Eq. 2); light (purple) are the expected layer from the cumulative scores (Eq. 4).

Test of syntax:
the order of words conveys important information.

- 사람이 고양이를 물었다
- 고양이가 사람을 물었다

mean very different things!

If models are genuinely learning syntax, they should know something about word order...

Natural Language Inference (NLI)

also known as recognizing textual entailment (RTE)

James Byron Dean refused to move without blue jeans
{entails, contradicts, neither}
James Dean didn't dance without pants

Entailment : premise가 hypothesis를 포함하는지
Contradiction : premise가 hypothesis와 모순되는지
Neutral : 그 외

How should a (humanlike) NLI model that's sensitive to word order behave?

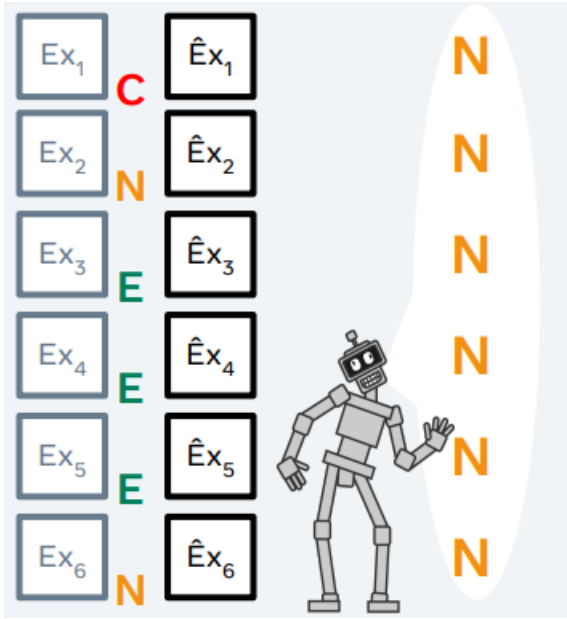
refused James jeans blue without Dean Byron move
to
{entails?, contradicts?, neither?}
didn't Dean James pants dance without

Fyodorov et al., 2000; Condoravdi et al., 2003; Bos and Markert, 2005; Dagan et al., 2006; MacCartney and Manning, 2009

(1) Maybe it just performs NLI

- For 3-way NLI, any pair that isn't clearly contradiction or entailment should be neutral .
- A model that learned this might just assign neutral always.

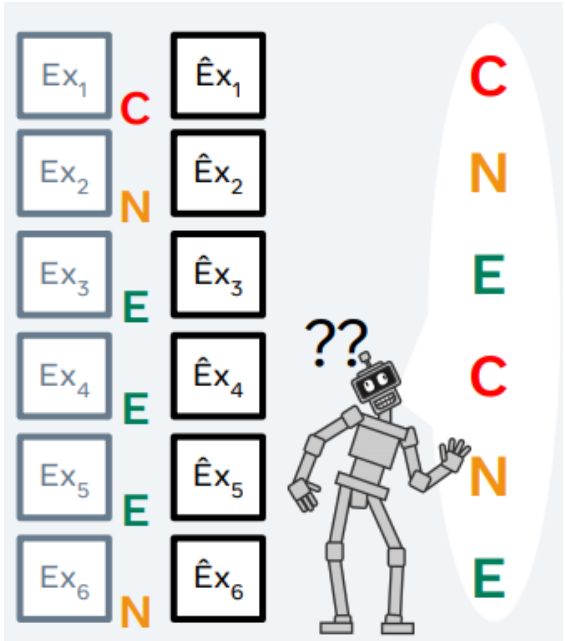
refused James jeans blue without Dean Byron move to
{entails?, contradicts?, neither?}
didn't Dean James pants dance without



(2) Maybe it will just be very uncertain…

- Perhaps it will just have no idea…then it should get roughly equal probability mass on all predictions.
- This is approximately the most frequent class baseline

refused James jeans blue without Dean Byron move to
{entails?, contradicts?, neither?}
didn't Dean James pants dance without



(3) Invariant to word order

- State-of-the-art NLI models are largely invariant to word order!
- Models often accept permuted examples (i.e. assign the original gold label to them).
- Same for pre-Transformer era neural models, too!

P: Boats in daily use lie within feet of the fashionable bars and restaurants .

H: There are boats close to bars and restaurants .

Premise	Hypothesis	Predicted Label
Boats in daily use lie within feet of the fashionable bars and restaurants.	There are boats close to bars and restaurants.	E
restaurants and use feet of fashionable lie the in Boats within bars daily .	bars restaurants are There and to close boats .	E
He and his associates weren't operating at the level of metaphor.	He and his associates were operating at the level of the metaphor.	C
his at and metaphor the of were He operating associates n't level .	his the and metaphor level the were He at associates operating of .	C

Constructing permutation function

- No word should appear in its original position
- A sentence of length n has (n-1)! possible permutations
- We select only unique permutations from this operation

- 100 permutation per example

- We are interested in **accuracy** on permuted sentences

Ex_1 = The cat sat on the mat → The cat was fat
...
Ex_6 = All cats are mammals → Felix is a mammal

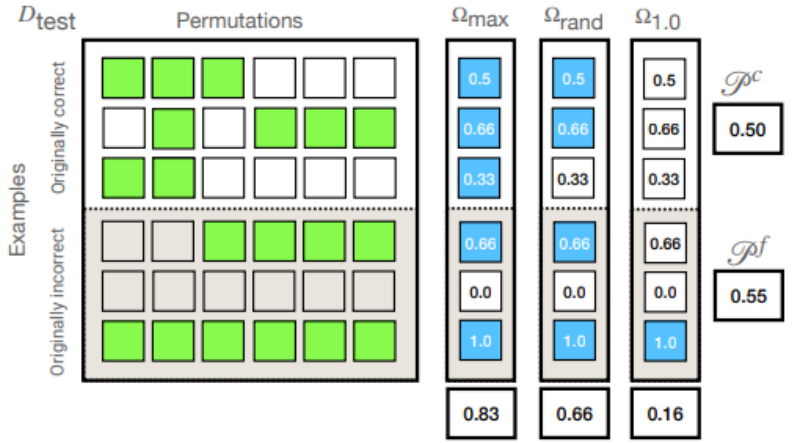
$\hat{Ex}_1$ = on sat The the mat cat → cat was The fat
...
$\hat{Ex}_6$ = are mammals cats All → Felix mammal a is

Experiment

- Trained models (RoBERTa, BART, DistilBERT, InferSent, ConvNet, BiLSTM) on MNLI to SOTA levels.
- Fine-tuned on (normal) MNLI.
- Evaluated on permuted MNLI, SNLI (in domain), ANLI (out of domain).

How many examples have at least one permutation predicting the gold label?

Model	Eval. Dataset	$\mathcal{A}$	Ω_{max}
RoBERTa-Large	MNLI_m_dev	0.906	0.987
	MNLI_mm_dev	0.901	0.987
	SNLI_dev	0.879	0.988
	SNLI_test	0.883	0.988
	A1*	0.456	0.897
	A2*	0.271	0.889
	A3*	0.268	0.902
	Mean	0.652	0.948
BART-Large	MNLI_m_dev	0.902	0.989
	MNLI_mm_dev	0.900	0.986
	SNLI_dev	0.886	0.991
	SNLI_test	0.888	0.990
	A1*	0.455	0.894
	A2*	0.316	0.887
	A3*	0.327	0.931
	Mean	**0.668**	**0.953**
DistilBERT	MNLI_m_dev	0.800	0.968
	MNLI_mm_dev	0.811	0.968
	SNLI_dev	0.732	0.956
	SNLI_test	0.738	0.950
	A1*	0.251	0.750
	A2*	0.300	0.760
	A3*	0.312	0.830
	Mean	0.564	0.883

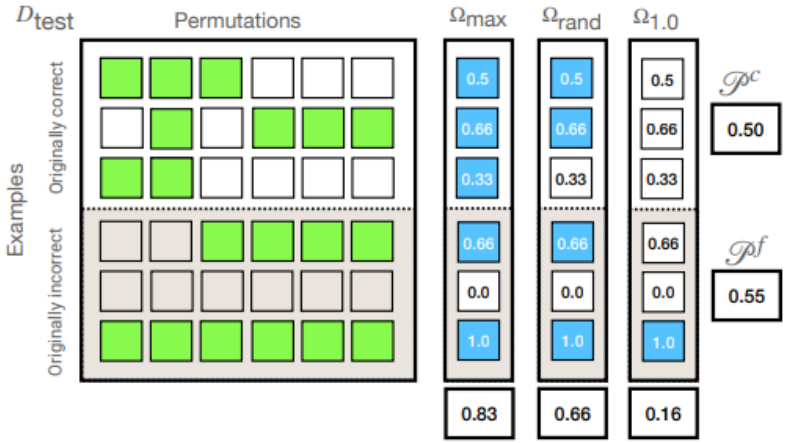


E_1: 3 gold label assignments (50%)
E_2: 3 gold label assignments (50%)
E_3: 2 gold label assignments (33%)
E_4: 4 gold label assignments (66%)
E_5: 0 gold label assignments ~~(00%)~~
E_6: 6 gold label assignments (100%)

Ω_{max} = % examples = 83%

How many examples have at least one permutation predicting the gold label?

Model	Eval. Dataset	$\mathcal{A}$	Ω_{rand}
RoBERTa-Large	MNLI_m_dev	0.906	0.794
	MNLI_mm_dev	0.901	0.790
	SNLI_dev	0.879	0.826
	SNLI_test	0.883	0.828
	A1*	0.456	0.364
	A2*	0.271	0.359
	A3*	0.268	0.397
	Mean	0.652	0.623
BART-Large	MNLI_m_dev	0.902	0.784
	MNLI_mm_dev	0.900	0.788
	SNLI_dev	0.886	0.834
	SNLI_test	0.888	0.836
	A1*	0.455	0.374
	A2*	0.316	0.397
	A3*	0.327	0.424
	Mean	**0.668**	**0.634**
DistilBERT	MNLI_m_dev	0.800	0.779
	MNLI_mm_dev	0.811	0.786
	SNLI_dev	0.732	0.731
	SNLI_test	0.738	0.725
	A1*	0.251	0.300
	A2*	0.300	0.343
	A3*	0.312	0.363
	Mean	0.564	0.575



E_1: 3 gold label assignments (50%)
E_2: 3 gold label assignments (50%)
E_3: 2 gold label assignments ~~(33%)~~
E_4: 4 gold label assignments (66%)
E_5: 0 gold label assignments ~~(00%)~~
E_6: 6 gold label assignments (100%)

Ω_{rand} = ⅔ examples = 63%

How many examples have ALL permutations predicting the gold label?

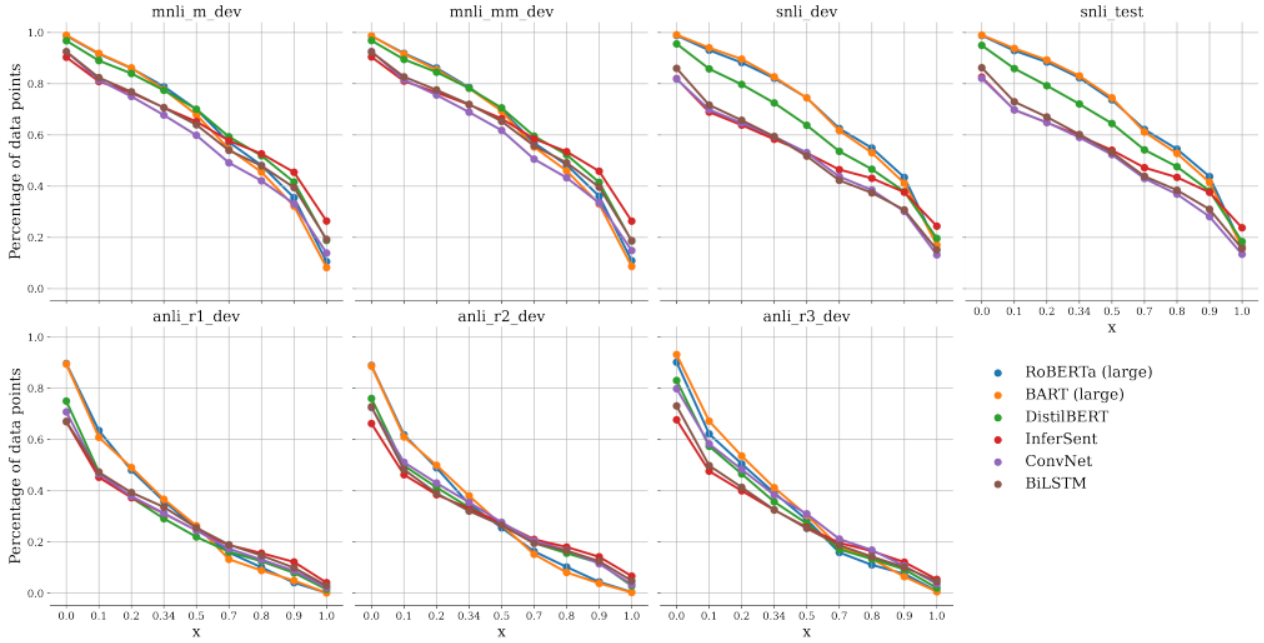
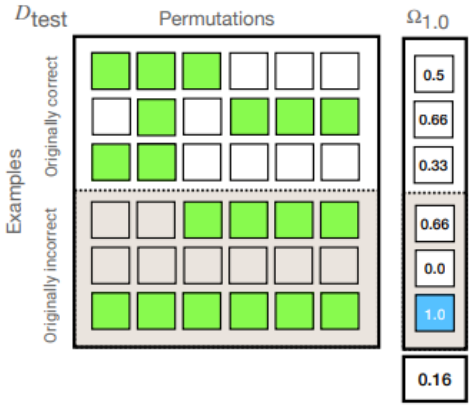


Figure 8: Ω_x threshold for all datasets with varying x and computing the percentage of examples that fall within the threshold. The top row consists of in-distribution datasets (MNLI, SNLI) and the bottom row contains out-of-distribution datasets (ANLI)



E_1: 3 gold label assignments (50%)
E_2: 3 gold label assignments (50%)
E_3: 2 gold label assignments ~~(33%)~~
E_4: 4 gold label assignments (66%)
E_5: 0 gold label assignments ~~(00%)~~
E_6: 6 gold label assignments (100%)

$\Omega_{1.0}$ = ⅙ examples = 16%

We observed that for some examples the models initially got wrong, there exists (a) permutation(s) that receive(s) the gold label!

P: Castlerigg near Keswick is the best example.
H: A good example would be Keswick near Castlerigg.

Correct label : Entailment
RoBERTa (large): Contradiction

P: best Castlerigg near example Keswick is the .
H: Keswick example near good Castlerigg be A would .

RoBERTa (large): Entailment

FLIPS: What percentage of permutations predict gold label, whose original pairs were INCORRECTLY predicted?

Model	Eval. Dataset	$\mathcal{P}^c$	$\mathcal{P}^f$
RoBERTa-Large	MNLI_m_dev	0.707	0.383
	MNLI_mm_dev	0.707	0.387
	SNLI_dev	0.768	0.393
	SNLI_test	0.760	0.407
	A1*	0.392	0.286
	A2*	0.465	0.292
	A3*	0.480	0.308
	Mean	0.611	**0.351**
BART-Large	MNLI_m_dev	0.689	0.393
	MNLI_mm_dev	0.695	0.399
	SNLI_dev	0.762	0.363
	SNLI_test	0.762	0.370
	A1*	0.379	0.295
	A2*	0.428	0.303
	A3*	0.428	0.333
	Mean	0.592	**0.351**
DistilBERT	MNLI_m_dev	0.775	0.343
	MNLI_mm_dev	0.775	0.346
	SNLI_dev	0.767	0.307
	SNLI_test	0.770	0.312
	A1*	0.511	0.267
	A2*	0.619	0.265
	A3*	0.559	0.259
	Mean	**0.682**	0.300

- Note: for a classic Bag-of-Words, Pc would be 100% and Pf would be 0%!

Is it just for Transformers?

- Weaker models, weaker effect.
- Pf for non-Transformers is approximately the same as for transformers.

- Both architectures are similarly bag-of-words-y (though no investigated model is a strict BOW).

Model	Eval. Dataset	$\mathcal{A}$	Ω_{max}	$\mathcal{P}^c$	$\mathcal{P}^f$	Ω_{rand}
InferSent	MNLI_m_dev	0.658	0.904	0.842	0.359	0.712
	MNLI_mm_dev	0.669	0.905	0.844	0.368	0.723
	SNLI_dev	0.556	0.820	0.821	0.323	0.587
	SNLI_test	0.560	0.826	0.824	0.321	0.600
	A1*	0.316	0.669	0.425	0.395	0.313
	A2*	0.310	0.662	0.689	0.249	0.330
	A3*	0.300	0.677	0.675	0.236	0.332
	Mean	**0.481**	0.780	0.731	**0.322**	0.514
ConvNet	MNLI_m_dev	0.631	0.926	0.773	0.340	0.684
	MNLI_mm_dev	0.640	0.926	0.782	0.343	0.694
	SNLI_dev	0.506	0.819	0.813	0.339	0.597
	SNLI_test	0.501	0.821	0.809	0.341	0.596
	A1*	0.271	0.708	0.648	0.218	0.316
	A2*	0.307	0.725	0.703	0.224	0.356
	A3*	0.306	0.798	0.688	0.234	0.388
	Mean	0.452	**0.817**	**0.745**	0.291	0.519
BiLSTM	MNLI_m_dev	0.662	0.925	0.800	0.351	0.711
	MNLI_mm_dev	0.681	0.924	0.809	0.344	0.724
	SNLI_dev	0.547	0.860	0.762	0.351	0.598
	SNLI_test	0.552	0.862	0.771	0.363	0.607
	A1*	0.262	0.671	0.648	0.271	0.340
	A2*	0.297	0.728	0.672	0.209	0.328
	A3*	0.304	0.731	0.656	0.219	0.331
	Mean	0.472	0.814	0.731	0.301	**0.520**

The labels must be chosen by chance?

- Unfortunately, no. The average entropy for Transformers is pretty low, suggesting overconfidence*!
- BART has the lowest entropy/highest confidence!
- Pre-Transformer models are somewhat better, but probably due to their lower capacity

*although miscalibration might also come into play.

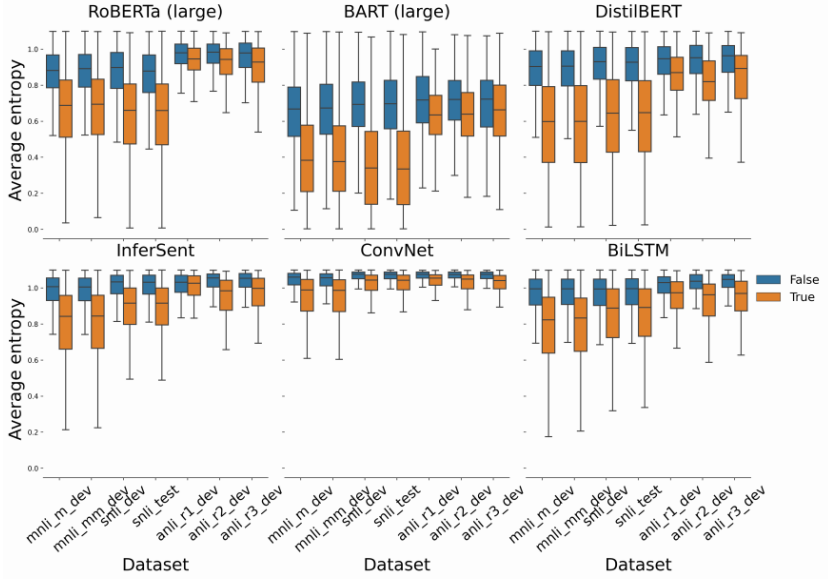
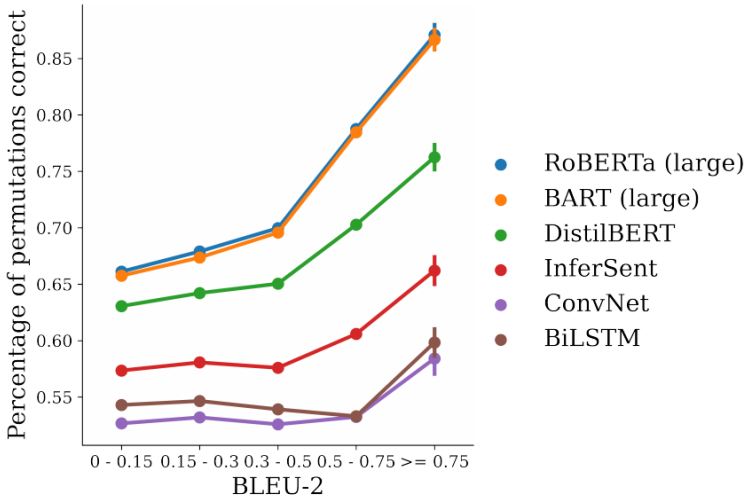


Figure 2: Average entropy of model confidences on permutations that yielded the correct results for Transformer-based models (top) and Non-Transformer-based models (bottom). Results are shown for D^c (orange) and D^f (blue). The boxes show the quartiles of the entropy distributions.

Which permutations do our models accept?

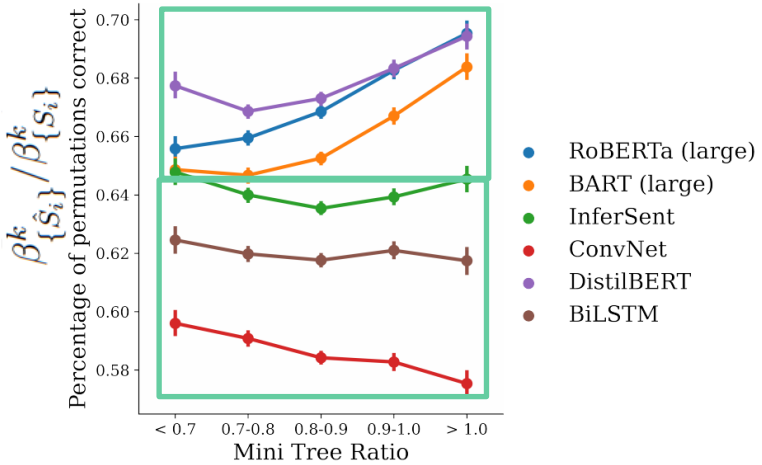
- Preserving local word order leads to accepted permutations
  - Percentage of permutations correct increases with more bi-gram overlap!
  - (BLEU-3 and BLEU-4 were too low to compare)



- Transformer LMs aren’t entirely BOW, they can handle some more abstract syntactic information

POS mini-tree overlap score:

$$\beta^k_{\{w_i,S_i\}} = \frac{1}{k} \mid \text{argmax}_k \psi^r_{\{w_i,D_{\text{train}}\}} \cap \text{argmax}_k \psi^r_{\{w_i,S_i\}} \mid$$



Human Analysis

Evaluator	Accuracy	Macro F1	Acc on D^c	Acc on D^f
X	0.581 ±0.068	0.454	0.649 ±0.102	0.515 ±0.089
Y	0.378 ±0.064	0.378	0.411 ±0.098	0.349 ±0.087

- 200 permuted sentences of varying length
- RoBERTa gets all of them “correct”!
- Annotators are “experts” in NLI

Once again, this time, in Chinese!

- Just to verify this, we looked into another language…
- Similar issue in Chinese OCNLI corpus!
- This isn't a tokenization complication, or some quirk of English.

Hu, Richardson, Xu, Li, Kuebler, Moss 2020 (EMNLP) **OCNLI**: Original Chinese Natural Language Inference

Model	$\mathcal{A}$	Ω_{max}	$\mathcal{P}^c$	$\mathcal{P}^f$	Ω_{rand}
RoBERTa-Large	**0.784**	**0.988**	0.726	**0.339**	**0.773**
InferSent	0.573	0.931	0.771	0.265	0.615
ConvNet	0.407	0.752	**0.808**	0.199	0.426
BiLSTM	0.566	0.963	0.701	0.271	0.611

Table 3: Results on evaluation on OCNLI Dev set. All models are trained on OCNLI corpus (Hu et al., 2020a). Bold marks the highest value per metric (**red** shows the model is insensitive to permutation).

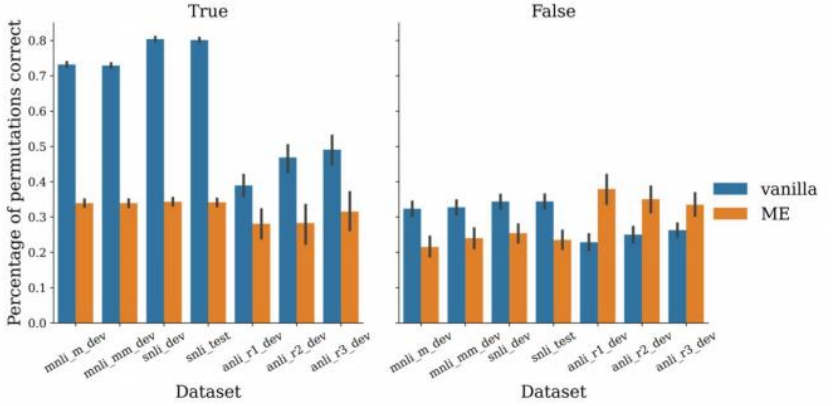
Initial Attempt: Max Entropy Training

A simple technique, but it works!

- Accuracy is constant while the percentage of accepted permutations reduced considerably!
- However, there's still room to improve!

* Similar approach concurrently by Gupta et al 2021

$$\mathcal{L} = \underset{\theta}{\operatorname{argmin}} \sum_{((p,h),y)} y \log(p(y|(p,h);\theta)) + \sum_{i=1}^{n} \mathbf{H}\left(y|(\hat{p}_i,\hat{h}_i);\theta\right)$$



Summary

1. All tested models are largely insensitive to permutations of word order, though humans are not.

2. Reordering words can cause models to flip classification labels

3. Models have learned some distributional information (POS neighborhood) that enable them to perform reasonably well under the permuted set up

References

- https://arxiv.org/pdf/2101.00010.pdf
- https://github.com/facebookresearch/UNLU
- https://www.cs.mcgill.ca/~ksinha4/unli/

Thanks