Natural Language Understanding, Generation, and Machine Translation

Lecture 18: Open-Vocabulary Models

Alexandra Birch

4 March 2022 (week 6)

School of Informatics
University of Edinburgh
a.birch@ed.ac.uk

Based on slides by Rico Sennrich

Refresher

Text Representation

how do we represent text?

- 1-hot encoding
  - lookup of word embedding for input
  - probability distribution over vocabulary for output
- large vocabularies
  - increase network size
  - decrease training and decoding speed
- typical network vocabulary size: 10 000–100 000 symbols

	representation of "cat"	
vocabulary	1-hot vector	embedding
$\begin{matrix}0\\1\\2\\.\\1024\end{matrix} \begin{bmatrix}\text{the}\\\text{cat}\\\text{is}\\.\\\text{mat}\end{bmatrix}$	$\begin{bmatrix}0\\1\\0\\.\\0\end{bmatrix}$	$\begin{bmatrix}0.1\\0.3\\0.7\\0.5\end{bmatrix}$

Problem

NLU and NLG are open-vocabulary problems

- many training corpora contain millions of word types
- productive word formation processes (compounding; derivation) allow formation and understanding of unseen words
- names, numbers are morphologically simple, but open word classes
- Rest of this class we are going to focus on **translation**

Open-vocabulary models

Non-Solution: Ignore Rare Words

- replace out-of-vocabulary words with UNK
- a vocabulary of 50 000 words covers 95% of text
- this gets you 95% of the way...
  ... if you only care about automatic metrics

why 95% is not enough
rare outcomes have high self-information

source	Mr **Gallagher** has offered a ray of hope.
reference	Herr **Gallagher** hat einen hoffnungsstrahl ausgesandt .

Solution 1: Approximative Softmax

approximative softmax [Jean et al., 2015]
compute softmax over "active" subset of vocabulary

$\rightarrow$ smaller weight matrix, faster softmax

- at training time: vocabulary based on words occurring in training set partition
- at test time: determine likely target words based on source text
  (using cheap method like translation dictionary)

limitations

- allows larger vocabulary, but still not open
- network may not learn good representation of rare words

Solution 2: Back-off Models

back-off models [Jean et al., 2015, Luong et al., 2015]

- replace rare words with UNK at training time
- when system produces UNK, align UNK to source word, and translate this with back-off method

source	The **indoor temperature** is very pleasant.	
reference	Das **Raumklima** ist sehr angenehm.	
[Bahdanau et al., 2015]	Die **UNK** ist sehr angenehm.	✗
[Jean et al., 2015]	Die **Innenpool** ist sehr angenehm.	✗

limitations

- compounds: hard to model 1-to-many relationships
- morphology: hard to predict inflection with back-off dictionary
- names: if alphabets differ, we need transliteration
- alignment: attention model unreliable

Solution 3: Subword NMT

Subwords for NMT: Motivation

Subwords units could be meaningful units of translation

- compounding and other productive morphological processes
  - they charge a carry-on bag fee.
  - sie erheben eine Hand|gepäck|gebühr.
- names
  - Obama(English; German)
  - Обама (Russian)
- technical terms, numbers, etc.:
  - 10-12-2020.
  - December 10 2020.

Subword units

segmentation algorithms: wishlist

- **open-vocabulary NMT**: encode *all* words through small vocabulary
- encoding generalizes to unseen words
- small text size
- good translation quality

our experiments [Sennrich et al., 2016]

- after preliminary experiments, we propose:
  - character n-grams (with shortlist of unsegmented words)
  - segmentation via *byte pair encoding* (BPE)

Byte pair encoding for word segmentation

bottom-up character merging

- starting point: character-level representation
  $\rightarrow$ computationally expensive
- compress representation based on information theory
  $\rightarrow$ byte pair encoding [Gage, 1994]
- repeatedly replace most frequent symbol pair ('A','B') with 'AB'
- hyperparameter: when to stop
  $\rightarrow$ controls vocabulary size

word	freq
'**l o** w</w>'	5
'**l o** w e r</w>'	2
'n e w **e s t</w>**'	6
'w i d **e s t</w>**'	3

vocabulary:

l o w</w> w e r</w> n s t</w> i d

es est</w> lo

Byte pair encoding for word segmentation

why BPE?

- open-vocabulary:
  operations learned on training set can be applied to unknown words
- compression of frequent character sequences improves efficiency
  $\rightarrow$ trade-off between text length and vocabulary size

'**l o** w **e s t</w>**'

e s	$\rightarrow$	**es**
es t</w>	$\rightarrow$	**est</w>**
l o	$\rightarrow$	**lo**

Evaluation: data and methods

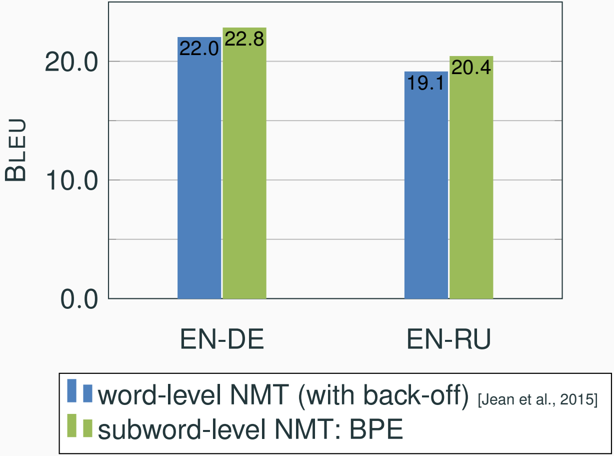
data

- WMT 15 English$\rightarrow$German and English$\rightarrow$Russian

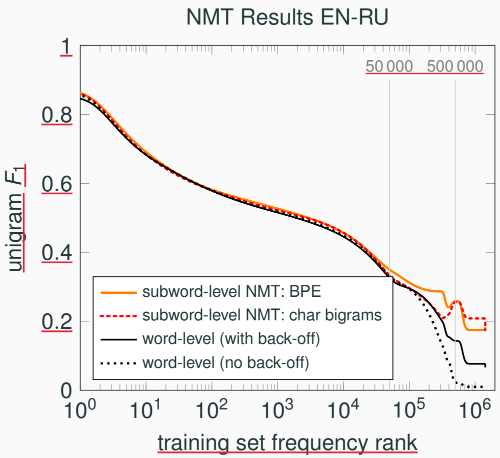
model

- attentional encoder–decoder neural network
- parameters and settings as in [Bahdanau et al, 2014]

Subword NMT: Translation Quality



Subword NMT: Translation Quality



Examples

system	sentence
source	health research institutes
reference	Gesundheitsforschungsinstitute
word-level (with back-off)	Forschungsinstitute
character bigrams	Fo|rs|ch|un|gs|in|st|it|ut|io|ne|n
BPE	Gesundheits|forsch|ungsin|stitute
source	rakfisk
reference	ракфиска (rakfiska)
word-level (with back-off)	rakfisk $\rightarrow$ UNK $\rightarrow$ rakfisk
character bigrams	ra|kf|is|k $\rightarrow$ ра|кф|ис|к (ra|kf|is|k)
BPE	rak|f|isk $\rightarrow$ рак|ф|иска (rak|f|iska)

Sharing BPE Vocabulary

- BPE merge operations for examples in previous slides are learned on concatenation of English and (romanized) Russian
- separate BPE can give inconsistent segmentations:
  rak|f|isk $\rightarrow$ пра|ф|иск (pra|f|isk)
- why? training data contains pair:
  p|rak|ri|ti$\rightarrow$пра|крит|и (pra|krit|i)
- with shared BPE, we get more consistent segmentation:
  pra|krit|i$\rightarrow$пра|крит|и (pra|krit|i)
- shared BPE has also proven useful for multilingual models

Subword Models: BPE-Dropout

- BPE-Dropout: Simple and effective Subword Regularizations [Provilkov et al., 2020]
- Adding stochastic noise to increase model robustness
- BPE: most frequent words are intact in vocabulary, learns how to compose with infrequent words
- If we sometimes forget to merge, we will learn how words compose, and better transliteration
- forget 1 in 10 times for most scripts, 6/10 in CKJ scripts
- Consistently give 1+ BLEU scores across language pairs - widely used

Subword Models: BPE-Dropout

```
u-n-r-e-l-a-t-e-d
u-n re-l-a-t-e-d
u-n re-l-at-e-d
u-n re-l-at-ed
un re-l-at-ed
un re-l-ated
un rel-ated
un-related
unrelated
```

(a)

```
u-n_r-e-l-a_t-e_d
u-n re-l_a-t-e_d
u-n re_l-at-e_d
un re-l-at-e-d
un re_l-at-ed
un re-lat-ed
un relat_ed
```

```
u-n-r-e-l-a_t-e-d
u_n re_l-a-t-e-d
u_n re-l-at-e-d
u_n re-l-ate_d
u_n rel-ate-d
u_n relate_d
```

```
u-n_r_e_l-a-t-e-d
u-n-r_e-l-at-e-d
u-n-r_e-l_at_ed
un-r-e-l-at-ed
un re-l_at-ed
un re-l-ated
un rel_ated
```

(b)

BPE

BPE dropout

From [Provilkov et al., 2020]

- Hyphen - possible merge
- merges performed - in green
- merges dropped - in red

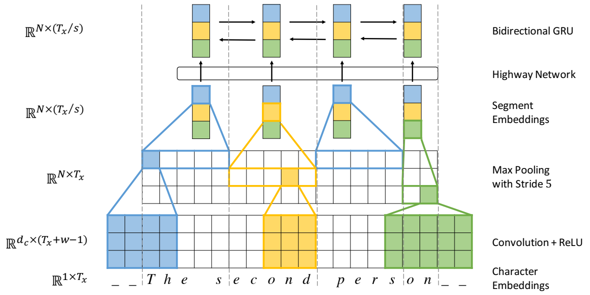
Solution 4: Character-level NMT

Character-level Models

- advantages:
  - (mostly) open-vocabulary
  - no heuristic or language-specific segmentation
  - neural network can conceivably learn from raw character sequences
- drawbacks:
  - increasing sequence length slows training/decoding (reported x2–x8 increase in training time)
- open questions
  - on which level should we represent meaning?
  - on which level should attention operate?

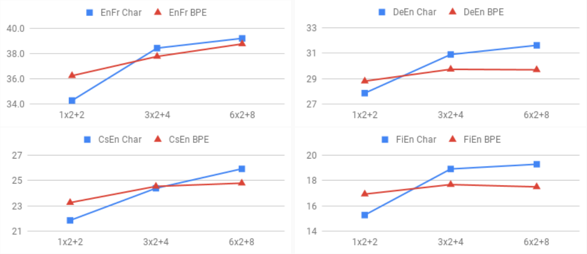
Fully Character-level NMT [Lee et al., 2016]

- goal: get rid of word boundaries
- source side: convolution and max-pooling layers
- character-level RNN on target side



Large-Capacity Character-level NMT [Cherry et al., 2018]

- train deep attentional LSTM encoder-decoders
- for shallow model, BPE is best quality
- for deep model, char-level model is better
- main problem for char-level: training time (8x slowdown)
- open challenge: compress representation without loss in quality



Beyond Character-level

- Massively multilingual settings character-level models can result in a very large vocabulary. eg. Unicode 143,859 codepoints
- Byte level input: better robustness to noise but longer training time ByT5: Towards a token-free future with pre-trained byte-to-byte models [Xue et al., 2021]
- Claim: token free - but really use fixed Unicode tokenisation which is not linguistically motivated
- Potentially unfair: Unicode characters beyond ASCII are much longer byte sequences - more expensive to model
- Pixel level: similarities that human readers might pick up on eg. to generalise to rare Chinese characters
- Makes translation significantly more robust to induced noise (including unicode errors) Robust Open-Vocabulary Translation from Visual Text Representations [Salesky et al., 2021]

Conclusion

- BPE and BPE-dropout is widely used
- There is no perfect method of handling tokenization.
- Opposing goals:
    - Decompose maximally for simple and robust processing
    - Desire to be computationally efficient in a way that is fair across languages
- Still not learning entities jointly with the rest of the model: separate preprocessing step
- How well these methods generalise from character strings to higher level of representation still to be fully studied

$\rightarrow$ next lecture: Pretrained language models and prompting

Bahdanau, D., Cho, K., and Bengio, Y. (2015).
Neural Machine Translation by Jointly Learning to Align and Translate.
In
Proceedings of the International Conference on Learning Representa

Cherry, C., Foster, G., Bapna, A., Firat, O., and Macherey, W. (2018).
Revisiting character-based neural machine translation with capacity and compression.
In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Chung, J., Cho, K., and Bengio, Y. (2016).
A Character-level Decoder without Explicit Segmentation for Neural Machine Translation.
CoRR, abs/1603.06147.

Gage, P. (1994).
A New Algorithm for Data Compression.
C Users J., 12(2):23–38.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015).
On Using Very Large Target Vocabulary for Neural Machine Translation.
In
Proceedings of the 53rd Annual Meeting of the Association for Comp

pages 1–10, Beijing, China. Association for Computational Linguistics.

Lee, J., Cho, K., and Hofmann, T. (2016).
Fully Character-Level Neural Machine Translation without Explicit Segmentation.
ArXiv e-prints.

Ling, W., Trancoso, I., Dyer, C., and Black, A. W. (2015).
Character-based Neural Machine Translation.
ArXiv e-prints.

Luong, M.-T. and Manning, D. C. (2016).
Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models.
In
Proceedings of the 54th Annual Meeting of the Association for Comp
pages 1054–1063. Association for Computational Linguistics.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., and Zaremba, W. (2015).
Addressing the Rare Word Problem in Neural Machine Translation.
In
Proceedings of the 53rd Annual Meeting of the Association for Comp

pages 11–19, Beijing, China. Association for Computational Linguistics.

Provilkov, I., Emelianenko, D., and Voita, E. (2020).
Bpe-dropout: Simple and effective subword regularization.
In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1882–1892.

Salesky, E., Etter, D., and Post, M. (2021).
Robust open-vocabulary translation from visual text representations.
In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7235–7252, Online

and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016).
Neural Machine Translation of Rare Words with Subword Units.
In
Proceedings of the 54th Annual Meeting of the Association for Compu
pages 1715–1725, Berlin, Germany.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. (2021).
Byt5: Towards a token-free future with pre-trained byte-to-byte models.