Machine Translation

07: Open-vocabulary Translation

Rico Sennrich

University of Edinburgh

Text Representation

how do we represent text in NMT?

- 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

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

Problem

translation is open-vocabulary problem

- 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

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

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	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.	✗
[**Sennrich**, Haddow, Birch, ACL 2016]	Die **Innen+ temperatur** ist sehr angenehm.	✓

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

Subwords for NMT: Motivation

MT is an open-vocabulary problem

- 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)
  - オバマ (o-ba-ma) (Japanese)
- technical terms, numbers, etc.

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
  → computationally expensive
- compress representation based on information theory
  → byte pair encoding [Gage, 1994]
- repeatedly replace most frequent symbol pair ('A','B') with 'AB'
- hyperparameter: when to stop
  → 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

Byte pair encoding for word segmentation

bottom-up character merging

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

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

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

Byte pair encoding for word segmentation

bottom-up character merging

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

word	freq
'l o w</w>'	5
'l o w e r</w>'	2
'n e w **est</w>**'	6
'w i d **est</w>**'	3

vocabulary:
l o w</w> w e r</w> n s t</w> i d
es **est</w>**

Byte pair encoding for word segmentation

bottom-up character merging

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

word	freq
'**lo** w</w>'	5
'**lo** w e r</w>'	2
'n e w **est</w>**'	6
'w i d **est</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

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
  → trade-off between text length and vocabulary size

'l o w **es** t</w>'

e s	→	**es**
es t</w>	→	est</w>
l o	→	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 **est</w>**'

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

'**lo** w est</w>'

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

Evaluation: data and methods

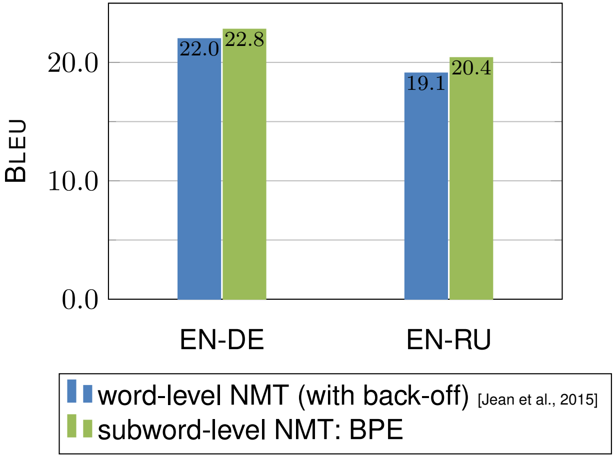
data

- WMT 15 English→German and English→Russian

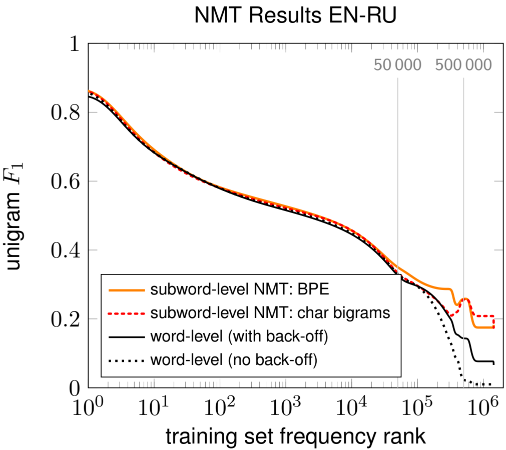
model

- 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)

1. Refresher
2. Open-vocabulary models
   - Non-Solution: Ignore Rare Words
   - Solution 1: Approximative Softmax
   - Solution 2: Back-off Models
   - Solution 3: Subword NMT
   - 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–x4 increase in training time)
  - naive char-level encoder-decoders are currently resource-limited [Luong and Manning, 2016]
- open questions
  - on which level should we represent meaning?
  - on which level should attention operate?

Character-level Models

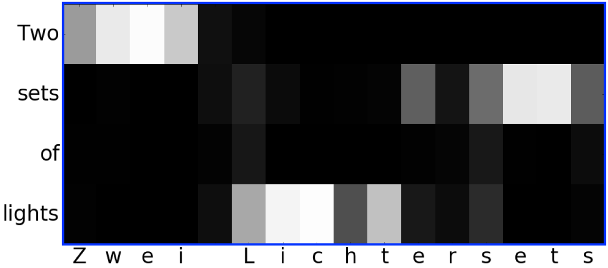
hierarchical model: back-off revisited [Luong and Manning, 2016]

- word-level model produces UNKs
- for each UNK, character-level model predicts word based on word hidden state
- pros:
  - prediction is more flexible than dictionary look-up
  - more efficient than pure character-level translation
- cons:
  - independence assumptions between main model and backoff model

Character-level Models

character-level output [Chung et al., 2016]

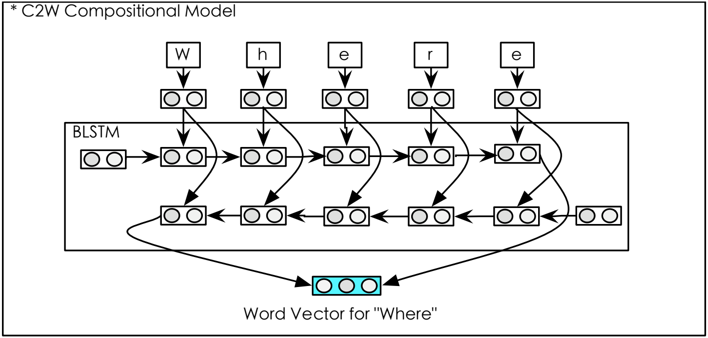
- no word segmentation on target side
- encoder is BPE-level
- good results for EN→{DE,CS,RU,FI}
- long training time (≈ x2 compared to BPE-level model)



Character-level Models

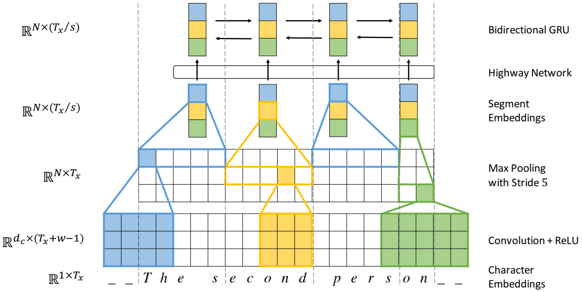
character-level input [Ling et al., 2015]

hierarchical representation: RNN states represent words, but their representation is computed from character-level LSTM



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

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



Conclusion

- BPE-level subword segmentation is currently the most widely used technique for open-vocabulary NMT
- character-level models are theoretically attractive, but currently require specialized architectures and more computational resources
- the presented methods allow open vocabulary; how well we generalize is other question
  $\rightarrow$ next lecture: morphology

Bibliography I

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 Representations (ICLR).

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 Computational Linguistics and the 7th International Joint Conference o
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.

Bibliography II

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 Computational Linguistics (Volume 1: Long Papers), 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 Computational Linguistics and the 7th International Joint Conference c
pages 11–19, Beijing, China. 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 Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany.