Natural Language Understanding, Generation, and Machine Translation

Lecture 6: Modelling Data and Words

Alexandra Birch

26 January 2023 (week 2)

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

Based on slides by Rico Sennrich

Refresher

Input Representation

how do we represent input?

- 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
$\begin{array}{c} 0 \\ 1 \\ 2 \\ . \\ 1024 \end{array} \begin{bmatrix} \text{the} \\ \text{cat} \\ \text{is} \\ . \\ \text{mat} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ . \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.1 \\ 0.3 \\ 0.7 \\ 0.5 \end{bmatrix}$

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

Problem

Research: we can download a clean corpus eg. Hansard, Europarl, Penn Treebank
Real life: nothing like this

- What if we need data that is not available publically? medical, financial, conversational
- What if it is for a low-resource language and none available?
- What if data we have is very noisy?
- What if there are very long dependencies over many sentences?

Modelling Data

Language Identification

Many datasets are crawls from the internet: CommonCrawl (250B pages) or the Internet Archive (850B pages)

- First step is Language Identification: LID
- What languages are these?
  - ndiyahamba (I'm going)
  - This is lekker bru (This is lovely)
- Generally solved task for high-resource languages
- Challenge with low resource, closely related language, code-switched or noisy data
- Solution: Fast lightweight classifiers
- Fasttext - Pre-trained models for 157 different languages

Sentence Splitting

Real data comes unsegmented into sentences.

- Sentence Segmentation:
  ! ? Mostly unabiguous but "." is very ambiguous (eg. the U.N.)
- Many scripts do not have end of sentence marker
- Speech is often not easy to segment into complete sentences
- Clean sentences - normally what models are trained on can struggle with shorter/longer sequences
- Solution: Language specific rules

Tokenisation

What is a word?

- Lookup in dictionary - but morphology makes this harder
- Thing between spaces - what about language without spaces or Finnish?
- Punctuation? Contractions? “that’s” $\rightarrow$ “that” “’s”
- Solution: For languages with spaces use spaces + punctuation + rules
- For Chinese etc. large dictionaries, punctuation + rules

Modelling Data

Critical for input to neural network - what is the input?
What sequence?

- Document, sentence, window, turn or utterance in a conversation

Sequence of what?

- Words, tokenized words, word stems, morphemes

Very long sequences are harder to model.
Vocabulary size needs to be limited as it has a huge effect on model size and efficiency.

Modelling words - 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

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

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

Subwords units could be meaningful useful for translation

- compounding and other productive morphological processes
  - they charge a carry-on bag fee.
  - sie erheben eine Hand|gepäck|gebühr.
- names
  - Edinburgh(English)
  - Edimburgo(Spanish)
- Morphological variation: slightly exaggerated eg. Turkish
  - OSMANLILAŞTIRAMAYABİLECEKLERİIMİZDENMİSŞINİZ
  - OSMAN-LI-LAŞ-TIR-AMA-YABİL-ECEK-LER-İIMİZ-DEN-MİS-ŞINİZ
- 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'	5
'l o w e r'	2
'n e w e s t'	6
'w i d e s t '	3

vocabulary:
l o w e r n s t 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'	5
'l o w e r'	2
'n e w **es** t'	6
'w i d **es** t '	3

vocabulary:

l o w e r n s t i d

es

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'	5
'l o w e r'	2
'n e w **est**'	6
'w i d **est** '	3

vocabulary:

l o w e r n s t i d

es **est**

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
'**lo** w'	5
'**lo** w e r'	2
'n e w **est**'	6
'w i d **est** '	3

vocabulary:
l o w e r n s t i d
es est **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’

e s	$\rightarrow$	es
es t	$\rightarrow$	est
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

’l o w **es** t’

e s	$\rightarrow$	**es**
es t	$\rightarrow$	est
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

'l o w **est**'

e s	$\rightarrow$	es
es t	$\rightarrow$	**est**
l o	$\rightarrow$	lo

Byte pair encoding for word segmentation

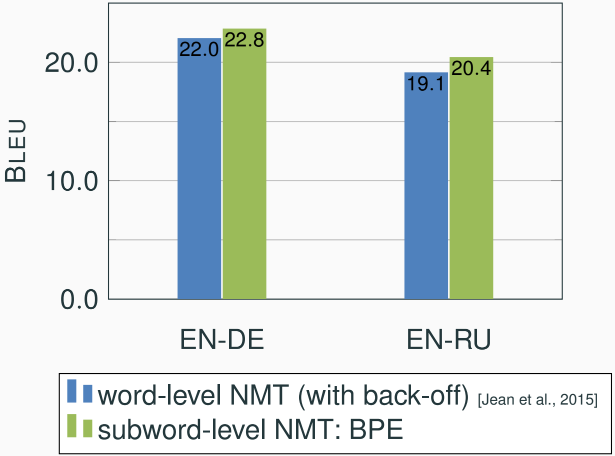
why BPE?

- 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'

e s	$\rightarrow$	es
es t	$\rightarrow$	est
l o	$\rightarrow$	**lo**

Subword NMT: Translation Quality



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

Subword Models: BPE-Dropout

- BPE-Dropout: Simple and effective Subword Regularizations

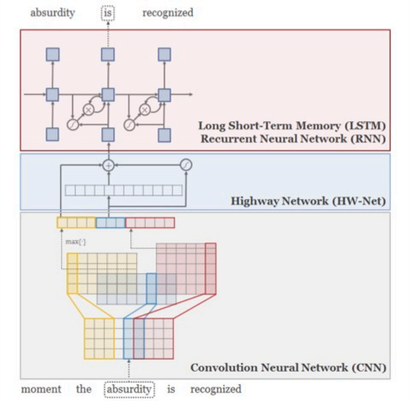
  [Provilkov et al., 2020]
- Adding stochastic noise to increase model robustness
- BPE: most frequent words are intact in vocabulary, learns how to compose with infrequent words
- If we sometimes forget to merge, we will learn how words compose, and better transliteration
- forget 1 in 10 times for most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

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?

Character Aware Neural Language Model [Kim et al., 2016]

- goal: vocabulary over character set
- Convolution over characters, highway network over words, and LSTM layers



Character Aware Neural Language Model [Kim et al., 2016]

(Based on cosine similarity)

	In Vocabulary				
	while	*his*	*you*	*richard*	*trading*
Word Embedding	*although*	*your*	*conservatives*	*jonathan*	*advertised*
	letting	*her*	*we*	*robert*	*advertising*
	though	*my*	*guys*	*neil*	*turnover*
	minute	*their*	*i*	*nancy*	*turnover*
Characters (before highway)	*chile*	*this*	*your*	*hard*	*heading*
	whole	*hhs*	*young*	*rich*	*training*
	meanwhile	*is*	*four*	*richer*	*reading*
	white	*has*	*youth*	*richter*	*leading*
Characters (after highway)	*meanwhile*	*hhs*	*we*	*eduard*	*trade*
	whole	*this*	*your*	*gerard*	*training*
	though	*their*	*doug*	*edward*	*traded*
	nevertheless	*your*	*i*	*carl*	*trader*

Beyond Character-level

- Massively multilingual settings character-level models can result in a very large vocabulary. eg. Unicode 1,112,064 codepoints
- Byte level:
  - 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

- Understand how your data was preprocessed
- Important to model it correctly
- 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

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

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.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. (2016).
Character-aware neural language models.
In Proceedings of the AAAI conference on artificial intelligence, volume 30.

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.