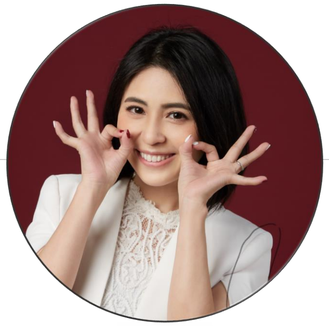
Applied Deep Learning



Tokenization

September 18th, 2024 http://adl.miulab.tw

National Taiwan University
國立臺灣大學

2 Vocabulary from Training Data

- Issue: unseen words cannot be well modeled (but human can)

	word		vocab mapping	embedding
Common words	hat	→	hat	
	learn	→	learn	
Variations	taaaaasty	→	UNK	
misspellings	laern	→	UNK	
novel items	Transformerify	→	UNK	

3 Morphological Typology

- Complex morphology, or word structure in many languages
  - E.g. Swahili verbs can have hundreds of conjugations, each encoding a wide variety of information. (tense, mood, definiteness, negation, information about the object, …)

Conjugation of *-ambia* [less ▲]

Non-finite forms		
Form	Positive	Negative
Infinitive	kuambia	kutoambia

Simple finite forms		
Positive form	Singular	Plural
Imperative	ambia	ambieni
Habitual	huambia	

Complex finite forms

Polarity	Persons				Persons / Classes		Classes											
	1st		2nd		3rd / M-wa		M-mi		Ma		Ki-vi		N		U	Ku	Pa	Mu
	Sg.	Pl.	Sg.	Pl.	Sg. / 1	Pl. / 2	3	4	5	6	7	8	9	10	11 / 14	15 / 17	16	18
Past [less ▲]																		
Positive	niliambia naliambia	tuliambia twaliambia	uliambia waliambia	mliambia mwaliambia	aliambia	waliambia	uliambia	iliambia	liliambia	yaliambia	kiliambia	viliambia	iliambia	ziliambia	uliambia	kuliambia	paliambia	muliambia
Negative	sikuambia	hatukuambia	hukuambia	hamkuambia	hakuambia	hawakuambia	haukuambia	haikuambia	halikuambia	hayakuambia	hakikuambia	havikuambia	haikuambia	hazikuambia	haukuambia	hakukuambia	hapakuambia	hamukuambia
Present [less ▲]																		
Positive	ninaambia naambia	tunaambia	unaambia	mnaambia	anaambia	wanaambia	unaambia	inaambia	linaambia	yanaambia	kinaambia	vinaambia	inaambia	zinaambia	unaambia	kunaambia	panaambia	munaambia
Negative	siambii	hatuambii	huambii	hamambii	haambii	hawaambii	hauambii	haiambii	haliambii	hayaambii	hakiambii	haviambii	haiambii	haziambii	hauambii	hakuambii	hapaambii	hamuambii
Future [less ▲]																		
Positive	nitaambia	tutaambia	utaambia	mtaambia	ataambia	wataambia	utaambia	itaambia	litaambia	yataambia	kitaambia	vitaambia	itaambia	zitaambia	utaambia	kutaambia	pataambia	mutaambia
Negative	sitaambia	hatutaambia	hutaambia	hamtaambia	hataambia	hawataambia	hautaambia	haitaambia	halitaambia	hayataambia	hakitaambia	havitaambia	haitaambia	hazitaambia	hautaambia	hakutaambia	hapataambia	hamutaambia
Subjunctive [less ▲]																		
Positive	niambie	tuambie	uambie	mambie	aambie	waambie	uambie	iambie	liambie	yaambie	kiambie	viambie	iambie	ziambie	uambie	kuambie	paambie	muambie
Negative	nisiambie	tusiambie	usiambie	msiambie	asiambie	wasiambie	usiambie	isiambie	lisiambie	yasiambie	kisiambie	visiambie	isiambie	zisiambie	usiambie	kusiambie	pasiambie	musiambie
Present conditional [less ▲]																		
Positive	ningeambia	tungeambia	ungeambia	mngeambia	angeambia	wangeambia	ungeambia	ingeambia	lingeambia	yangeambia	kingeambia	vingeambia	ingeambia	zingeambia	ungeambia	kungeambia	pangeambia	mungeambia
Negative	nisingeambia singeambia	tusingeambia hatungeambia	usingeambia hungeambia	msingeambia hamngeambia	asingeambia hangeambia	wasingeambia hawangeambia	usingeambia haungeambia	isingeambia haingeambia	lisingeambia halingeambia	yasingeambia hayangeambia	kisingeambia hakingeambia	visingeambia havingeambia	isingeambia haingeambia	zisingeambia hazingeambia	usingeambia haungeambia	kusingeambia hakungeambia	pasingeambia hapangeambia	musingeambia hamungeambia
Past conditional [less ▲]																		
Positive	ningaliambia	tungaliambia	ungaliambia	mngaliambia	angaliambia	wangaliambia	ungaliambia	ingaliambia	lingaliambia	yangaliambia	kingaliambia	vingaliambia	ingaliambia	zingaliambia	ungaliambia	kungaliambia	pangaliambia	mungaliambia
Negative	nisingaliambia singaliambia	tusingaliambia hatungaliambia	usingaliambia hungaliambia	msingaliambia hamngaliambia	asingaliambia hangaliambia	wasingaliambia hawangaliambia	usingaliambia haungaliambia	isingaliambia haingaliambia	lisingaliambia halingaliambia	yasingaliambia hayangaliambia	kisingaliambia hakingaliambia	visingaliambia havingaliambia	isingaliambia haingaliambia	zisingaliambia hazingaliambia	usingaliambia haungaliambia	kusingaliambia hakungaliambia	pasingaliambia hapangaliambia	musingaliambia hamungaliambia

4 Token Definition

- Character
  - Pros: no unseen, small vocab
  - Cons: semantics of multiple characters is difficult to model
- Subword (parts of words)
  - A dominant modern paradigm
  - A balance between word and character

Byte-Pair Encoding (BPE)

- BPE is a simple, effective strategy for defining a subword vocabulary
- The most common pair of consecutive bytes of data is replaced with a byte that does not occur within that data.
  1) Start with a vocabulary containing only characters and an “end-of-word” symbol.
  2) Using a corpus of text, find the most common pair of adjacent characters “a,b”; add subword “ab” to the vocab.
  3) Replace instances of the character pair with the new subword; repeat until desired vocab size

A demo video: https://youtu.be/HEikzVL-IZU

Byte-Pair Encoding (BPE) Demonstration

1) Start with a vocabulary containing only characters and an "end-of-word" symbol.

```
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
```

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

2) Using a corpus of text, find the most common pair of adjacent characters “a,b”; add subword “ab” to the vocab.

seen 7 times

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

seen 9 times seen 9 times

Choose One

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

3) Replace instances of the character pair with the new subword; repeat until desired vocab size

```
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
```

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

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

seen 9 times

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

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

seen 9 times

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

seen 7 times

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

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

seen 7 times

lo w </w>	: 5
lo w e r </w>	: 2
n e w est</w>	: 6
w i d est</w>	: 3

VOCAB

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

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

low </w>	: 5
low e r </w>	: 2
n e w est</w>	: 6
w i d est</w>	: 3

VOCAB

</w>	d	
e	i	l
n	o	r
s	t	w
es	est	est</w>
lo	low	**ne**

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

low </w>	: 5
low e r </w>	: 2
ne w est</w>	: 6
w i d est</w>	: 3

VOCAB

</w>	d	
e	i	l
n	o	r
s	t	w
es	est	est</w>
lo	low	ne **new**

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

low </w>	: 5
low e r </w>	: 2
new est</w>	: 6
w i d est</w>	: 3

VOCAB

</w>	d	
e	i	l
n	o	r
s	t	w
es	est	est</w>
lo	low	ne new
newest</w>		

Byte-Pair Encoding (BPE) Demonstration

2) Add the most common adjacent characters to the vocab.
3) Replace the character pairs with the new subword

low </w>	: 5
low e r </w>	: 2
newest</w>	: 6
w i d est</w>	: 3

VOCAB

</w>	d	
e	i	l
n	o	r
s	t	w
es	est	est</w>
lo	low	ne new
newest</w>		
low</w>		

Byte-Pair Encoding (BPE) Demonstration

- MERGES
  - e + s => es
  - es + t => est
  - est + </w> => est</w>
  - l + o => lo
  - lo + w => low
  - n + e => ne
  - ne + w => new
  - new + est</w> => newest</w>
  - low + </w> => low</w>

```
    </w>  d
    e  i  l
    n  o  r
    s  t  w
es est est</w>
lo low ne new
  newest</w>
   low</w>
```

Byte-Pair Encoding (BPE) Demonstration

- Handling unseen tokens: lowest `low est</w>`
  - `e + s => es`
  - `es + t => est`
  - `est + </w> => est</w>`
  - `l + o => lo`
  - `lo + w => low`
  - `n + e => ne`
  - `ne + w => new`
  - `new + est</w> => newest</w>`
  - `low + </w> => low</w>`

Byte-Pair Encoding (BPE) Demonstration

- Handling unseen tokens: powest <unk> o w est</w>
  - e + s => es
  - es + t => est
  - est + </w> => est</w>
  - l + o => lo
  - lo + w => low
  - n + e => ne
  - ne + w => new
  - new + est</w> => newest</w>
  - low + </w> => low</w>

BPE Properties

- Usually include frequent words and frequent subwords
  - Are often morphemes (e.g. –*est* or –*er*)
- A morpheme is the smallest meaning-bearing unit of a language
  - *unlikeliest* => *un-*, *likely*, -*est* (3 morphemes)

Multilingual BPE

- Multilingual models tokenize all language by a unified BPE

GPT-3

Working on NLP is fun, but tokenization is not fun.

Clear Show example

Tokens	Characters
14	51

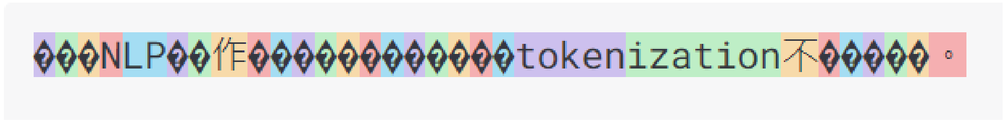
Working on NLP is fun, but tokenization is not fun.

GPT-3

做NLP工作很有趣，但tokenization不有趣。

Clear Show example

Tokens	Characters
29	27



Note: Your input contained one or more unicode characters that map to multiple tokens.

- Tokenizing Mandarin via Unicode encoding is inefficient → higher cost

https://platform.openai.com/tokenizer

Concluding Remarks

- Subword modeling addresses issues about unseen words
- Byte-pair encoding (BPE) is a commonly used method for subword tokenization
  - Include both frequent words and subwords (smallest meanining-bearing units)
- Different languages may need their own tokenization for better efficiency and lower cost