UniCase - Rethinking Casing in Language Models

Rafał Powalski[1] and **Tomasz Stanisławek[1,2]**

[1]Applica.ai

[2]Faculty of Mathematics and Information Science, Warsaw University of Technology

Mi2 DataLab seminar, 26.10.2020

Presentation plan

- Introduction
  - Basics
  - SOTA (BPE, Unigram)
  - Problems
- Proposed solution
  - General idea
  - Side effects
  - Experiments
  - Future work
- Useful links and references
- Discussion

Introduction

Text segmentation

Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. The term applies both to mental processes used by humans when reading text, and to artificial processes implemented in computers.

Introduction

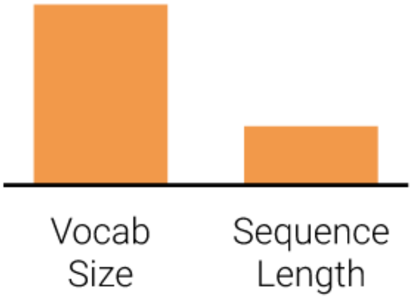
Tokenization - how machines read

Tokenization is a way of separating a piece of text into smaller units called tokens.

Introduction

Tokenization methods

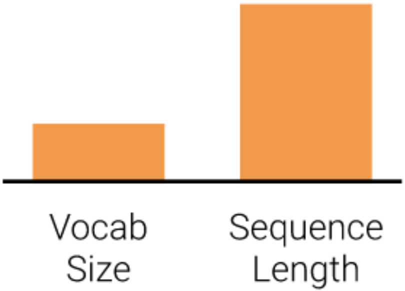
- Word tokens (word2vec, ...)



Introduction

Tokenization methods

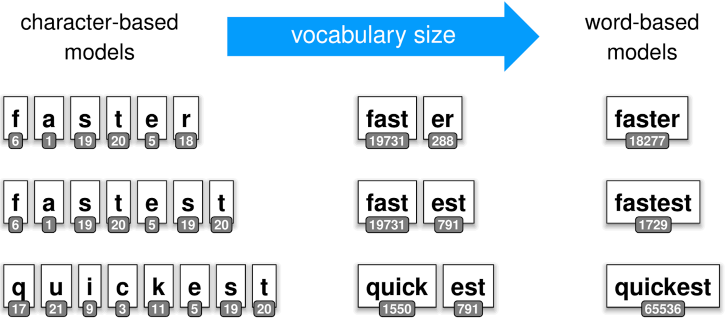
- Word tokens (word2vec, ...)
- Character tokens (Flair, ...)



Introduction

Tokenization methods

- Word tokens (word2vec, ...)
- Character tokens (Flair, ...)
- Sub-word tokens (BERT, RoBERTa, ...)



SOTA

Byte Pair Encoding (BPE)

Just uses the frequency of occurrences to identify the best match at every iteration until it reaches the predefined vocabulary size.

Unigram Subword Tokenization

A fully probabilistic model which does not use frequency occurrences. Instead, it trains a LM using a probabilistic model, removing the token which improves the overall likelihood the least and then starting over until it reaches the final token limit.

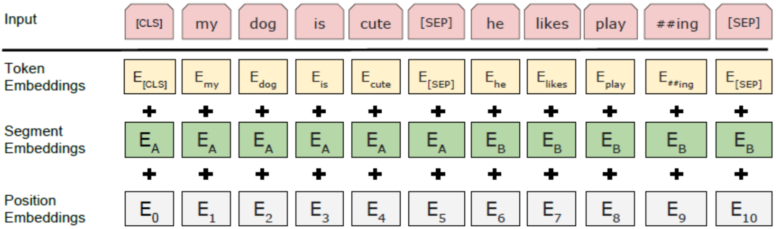
SOTA

Byte Pair Encoding is Suboptimal for Language Model Pretraining

Model	SQuAD 1.1 (dev.)		MNLI (dev.)		CoNLL NER	
	EM	F1	Acc. (m)	Acc. (mm)	Dev. F1	Test F1
Ours, BPE	80.6	88.2	81.4	82.4	94.0	90.2
Ours, Unigram LM	81.8	89.3	82.8	82.9	94.3	90.4
$\text{BERT}_{\text{BASE}}$	80.5	88.5	84.6	83.4	96.4	92.4

Table 3: Fine-tuning results. Metrics are averaged across 5 fine-tuning seeds; due to computational constraints we did not pretrain more than once per tokenization. We include fine-tuning results for a transformer with a comparable architecture, $\text{BERT}_{\text{BASE}}$, for reference, although we note that a direct comparison cannot be made due to $\text{BERT}_{\text{BASE}}$ using both a larger pretraining corpus and a larger subword vocabulary.

BERT



Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$
E_{my}
E_{dog}
E_{is}
E_{cute}
$E_{[SEP]}$
E_{he}
E_{likes}
E_{play}
$E_{\#\#ing}$
$E_{[SEP]}$
Segment Embeddings
E_A
E_A
E_A
E_A
E_A
E_A
E_B
E_B
E_B
E_B
E_B
Position Embeddings
E_0
E_1
E_2
E_3
E_4
E_5
E_6
E_7
E_8
E_9
E_{10}

Problems

1. Cased vs Uncased
2. With cased model subtokens semantics are different depends on capitalization
3. With cased model we need longer subtokens list for represent the same text sequence (we could pack less information to single span)

Problems

Examples with Roberta tokenizer

1. ‘ĠiPhone‘, ‘Ġi‘, ‘Phone‘
2. ‘ĠOTHER‘, ‘Ġother‘, ‘ĠOther‘
3. ‘ĠMc‘, ‘ĠMcC‘, ‘ĠMcDonald‘
4. ‘Acknowledgement‘ can be tokenize different depends on capitalization:
    - Title: ['ĠA', 'cknowled', 'gement']
    - Lower: ['Ġacknowledgement']
    - Upper: ['ĠAC', 'KN', 'OW', 'LED', 'G', 'EMENT']

Problems

Demo

Solution - tokenization

- Three main Case Shapes were choosen: Upper Case (XXX), Title Case (Xxx) and Lower Case (xxx)
- Create tokenizer which have coresponding tokens in chosen Case Shapes
- Tokenizer should split tokens identically for text with different casing.
- We chose Sentencepiece Unigram tokenizer and modified it to fulfil above conditions

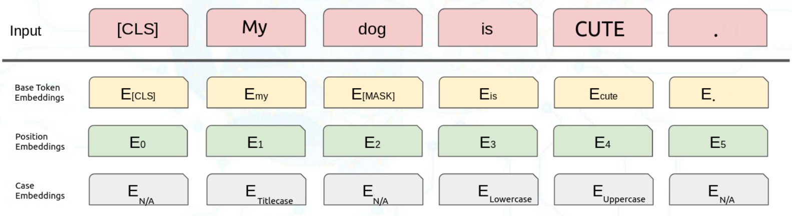
Solution - tokenization

How to create such thing?

- Create SPM tokenizer based on lowercased corpora
- Identify tokens which contains letters
- Modify protobuf of SPM model and add coresponding tokens with XXX and Xxx shape

Solution - model

- We want to use the same semantic embedding for all 3 Case shapes
- Shape information will be added as separate embedding containing learnable vectors for 3 shapes
- Model is trained with 2 training tasks: Base token prediction & Case prediction



Solution - examples

'Acknowledgement McDonald Other iPhone'

Simple words - original case on the beginning of the sentence

1. **BPE-Roberta**: '_Acknowled gement _McDonald _Other _iPhone'
2. **Unigram**: '_Ac knowledge ment _McDonald _Other _iPhone'
3. **UniCase**: '_Acknowledgement _Mc Donald _Other _i Phone'

Solution - examples

Simple words - lowercased

1. **BPE-Roberta**: '_acknowledgement _mc donald _other _iph one'
2. **Unigram**: '_acknowledgement _m c don ald _other _iphone'
3. **UniCase**: '_acknowledgement _mcdonald _other _i phone'

Solution - examples

Simple words - upper case

1. **BPE-Roberta**: '_AC KN OW LED G EMENT _M CD ON ALD _OTHER _IP H ONE'
2. **Unigram**: '_A CK NO W LED GE MENT _MC DO NA LD _OTHER _I PH ONE'
3. **UniCase**: '_ACKNOWLEDGEMENT _MCDONALD _OTHER _IPHONE'

Solution - results on GLUE

We have trained two separate models for the same no. of updates (125k, bs=2048):

- Roberta with Unigram tokenizer ($vocab\_size = 32k$)
- UniCase Model based on Unigram tokenizer ($vocab\_size = 32k$, $effective\_vocab\_size \approx 90k$)

Solution - results on GLUE

Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST	STS-B	Average
				Original casing					
UC	**58.29**	**85.18**	**90.88**	**91.29**	**88.16**	**71.84**	92.78	**88.18**	**83.29**
RB	57.92	84.84	90.33	91.01	88.14	69.31	**94.04**	87.80	82.87
				All texts from train and development sets were lowercased					
UC	**55.99**	**85.23**	**90.85**	**90.90**	88.13	**70.40**	92.89	**88.26**	**82.80**
RB	55.08	84.82	90.65	90.31	**88.14**	67.87	**94.15**	87.70	82.31
				All texts from train and development sets were uppercased					
UC	**56.25**	**85.19**	**91.28**	**91.10**	**88.09**	**71.84**	**92.83**	**88.11**	**83.07**
RB	39.24	80.11	87.84	87.23	86.90	62.82	89.11	85.21	77.19

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)
2. Experiments on NER problems (CoNLL, W-NUT)

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)
2. Experiments on NER problems (CoNLL, W-NUT)
3. Experiments on other languages (where capitalization is a problem)

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)
2. Experiments on NER problems (CoNLL, W-NUT)
3. Experiments on other languages (where capitalization is a problem)
4. Experiments on business usecases, where there are more upper cased texts

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)
2. Experiments on NER problems (CoNLL, W-NUT)
3. Experiments on other languages (where capitalization is a problem)
4. Experiments on business usecases, where there are more upper cased texts
5. Measure time for each experiment

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)
2. Experiments on NER problems (CoNLL, W-NUT)
3. Experiments on other languages (where capitalization is a problem)
4. Experiments on business usecases, where there are more upper cased texts
5. Measure time for each experiment
6. UniCase parameter optimization + ablation studies

Future work

1. Experiments on problems with longer text (WikiHop, TriviaQA, HotpotQA, OntoNotes, IMDB, Hyperpartisan)
2. Experiments on NER problems (CoNLL, W-NUT)
3. Experiments on other languages (where capitalization is a problem)
4. Experiments on business usecases, where there are more upper cased texts
5. Measure time for each experiment
6. UniCase parameter optimization + ablation studies
7. UniCase vs other techniques from Neural Machine Translation

Useful links and references

- UniCase - Rethinking Casing in Language Models
- (BPE) Neural Machine Translation of Rare Words with Subword Units
- (Unigram LM) Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates
- Byte Pair Encoding is Suboptimal for Language Model Pretraining
- Case-Sensitive Neural Machine Translation
- To Case or not to case: Evaluating Casing Methods for Neural Machine Translation
- RoBERTa: A Robustly Optimized BERT Pretraining Approach

Useful links and references

- https://blog.floydhub.com/tokenization-nlp/
- https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/
- https://colab.research.google.com/github/huggingface/transformers/
training-tokenizers.ipynb

Questions?

Thank you!