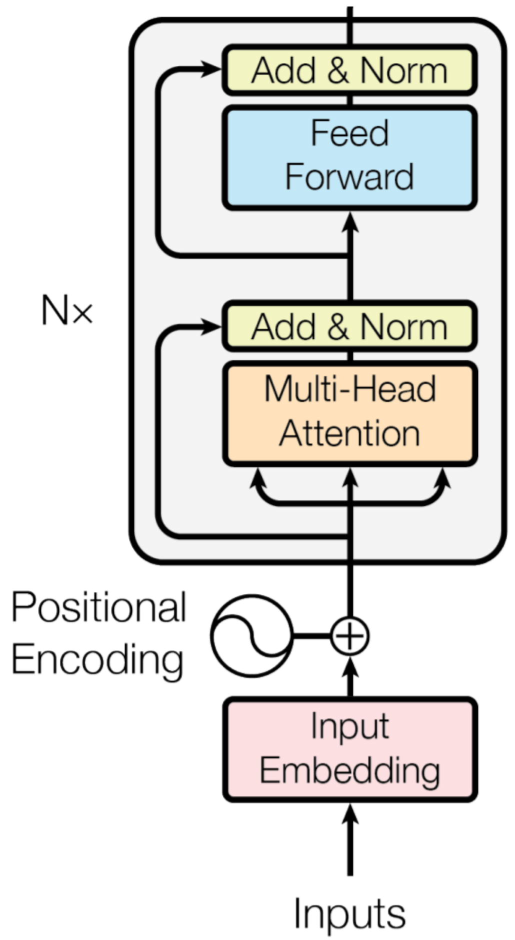
NATURAL LANGUAGE PROCESSING

LECTURE 8: Tokenization

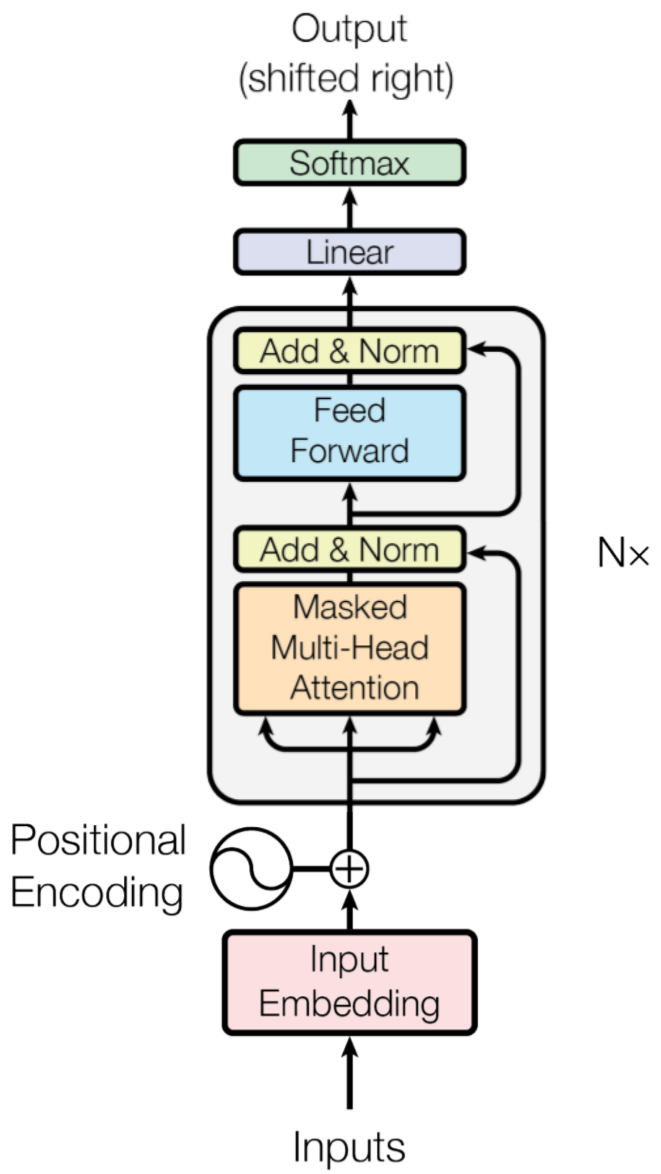
NLP Pipeline

- Data Collection
- Preprocessing
  - Pre-tokenization
  - Tokenization
- Modeling
- Training
  - Training Monitoring
- Evaluation
- Deployment
- Performance Monitoring

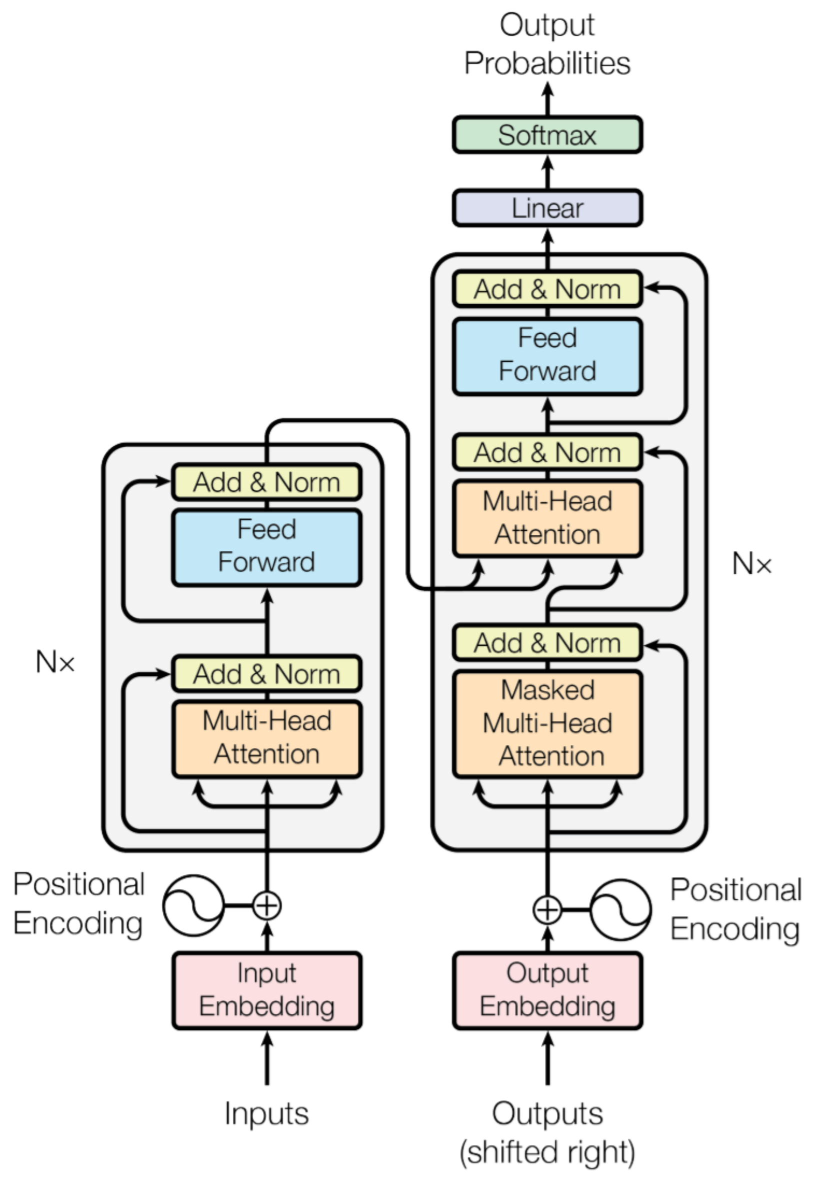
Encoder Pipeline



Decoder Pipeline

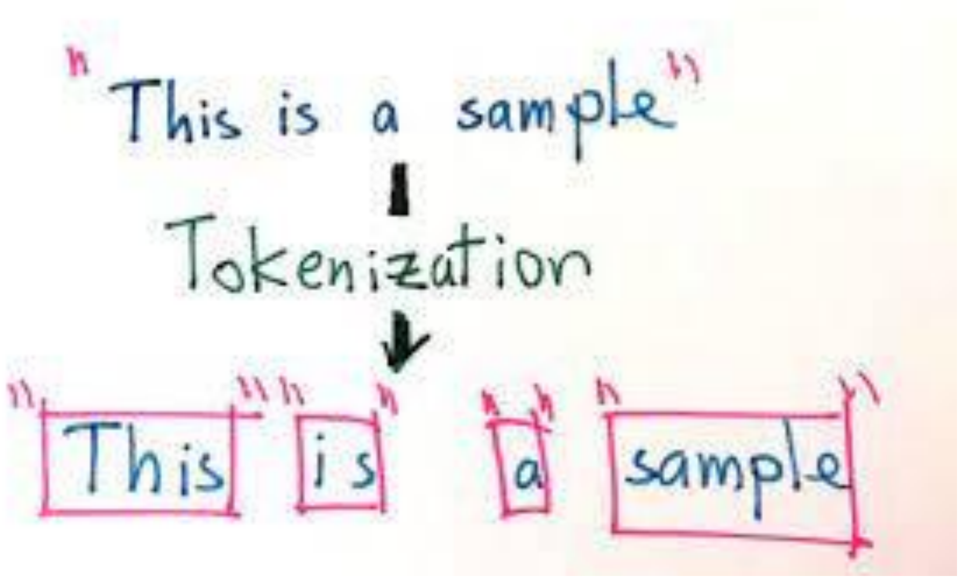


Sequence to sequence Pipeline



Tokenization

- Tokenization is a way of separating a piece of text into smaller units called tokens
- Based on the way of tokenization, out model recognize a sequence



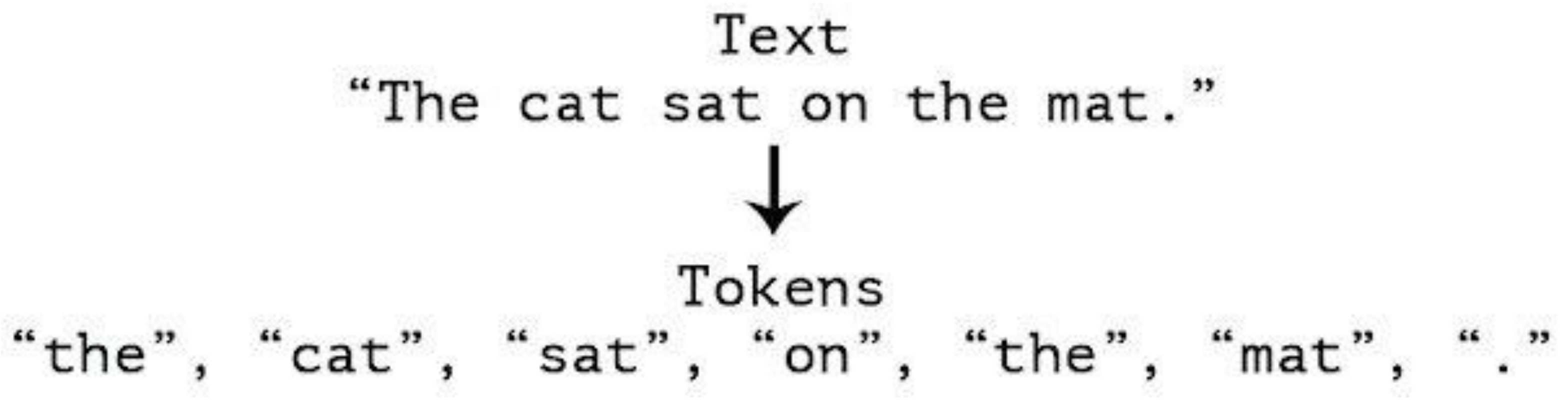
Tokenizer

- Tokenizers
  - Word-based Tokenizer
  - Character-based Tokenizer
  - Subword-based Tokenizer

- There can be a pre-tokenization step (e.g., space tokenization, Moses, Spacy, ftfy)

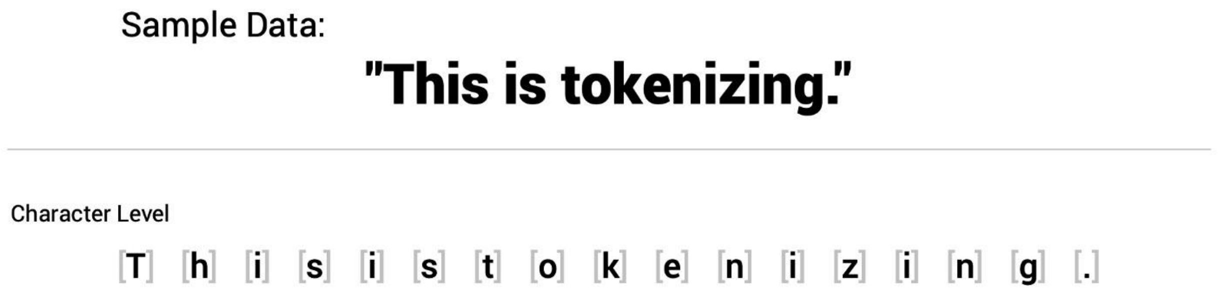
Word-based Tokenizer

- Input – [‘The devil is in the details’]
- Output – [‘The’, ‘devil’, ‘is’, ‘in’, ‘the’, ‘details’]


Text
“The cat sat on the mat.”
Tokens
“the”, “cat”, “sat”, “on”, “the”, “mat”, “.”

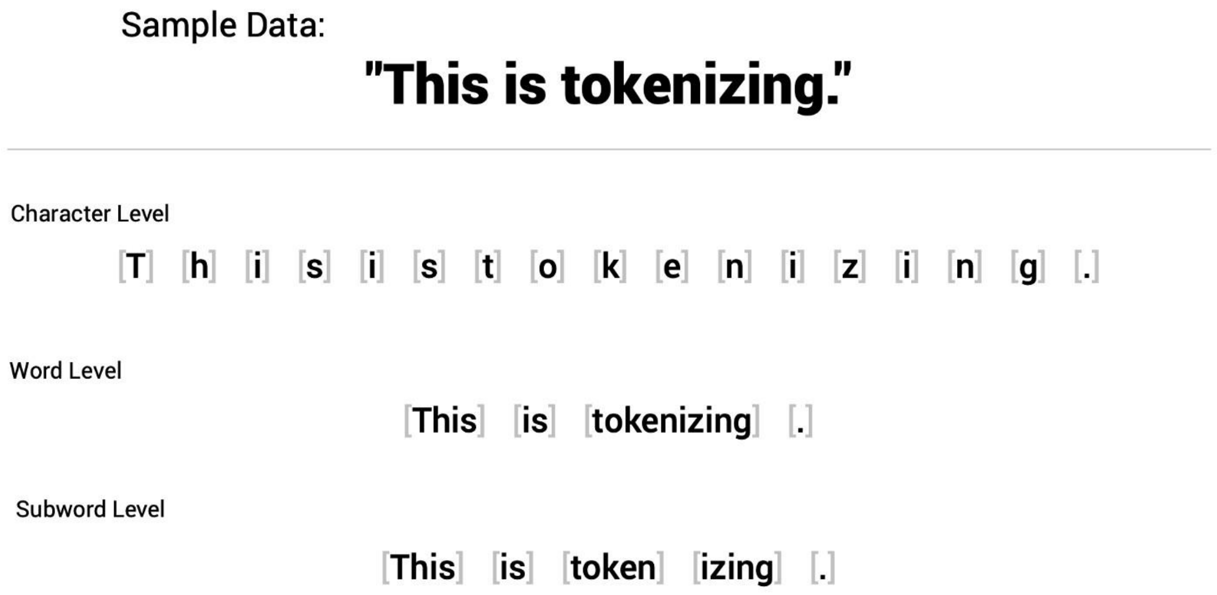
Character-based Tokenizer

- Input – [‘The devil is in the details’]
- Output – [‘T’, ‘h’, ‘e’, ‘d’, ‘e’, ‘v’, ‘i’, ‘l’, ‘i’, ‘s’, ‘i’, ‘n’, ‘t’, ‘h’, ‘e’, ‘d’, ‘e’, ‘t’, ‘a’, ‘i’, ‘l’, 's’]



Subword-based Tokenizer

- Input – [‘The devil is in the details’]
- Output – [‘The’, ‘de’, ‘vil’, ‘is’, ‘in’, ‘the’, ‘de’, ‘tail’, ‘s’]


Sample Data:
"This is tokenizing."
Character Level
[T] [h] [i] [s] [i] [s] [t] [o] [k] [e] [n] [i] [z] [i] [n] [g] [.]
Word Level
[This] [is] [tokenizing] [.]
Subword Level
[This] [is] [token] [izing] [.]

Pre-tokenization Recap

- Why should we set the pre-tokenization step?
  - Example
    - 이런 영화에 평점 1점 날리는 것들은 영화를 볼 줄 모르는 것들이다.
    - 어렸을때부터mbc드라마를봤었는데이제더이상은안봐야할것같다!막장스토리출생의비밀뒤바뀐운명자주등장하는대기업그속에서그려지는억지설정도모자라이젠쌍둥이등장?...
    - I got up early. Then I went to DAVIAN Lab.
  - Korean
    - 조사
  - English
    - Case, uncased

Word vs Character vs Subword

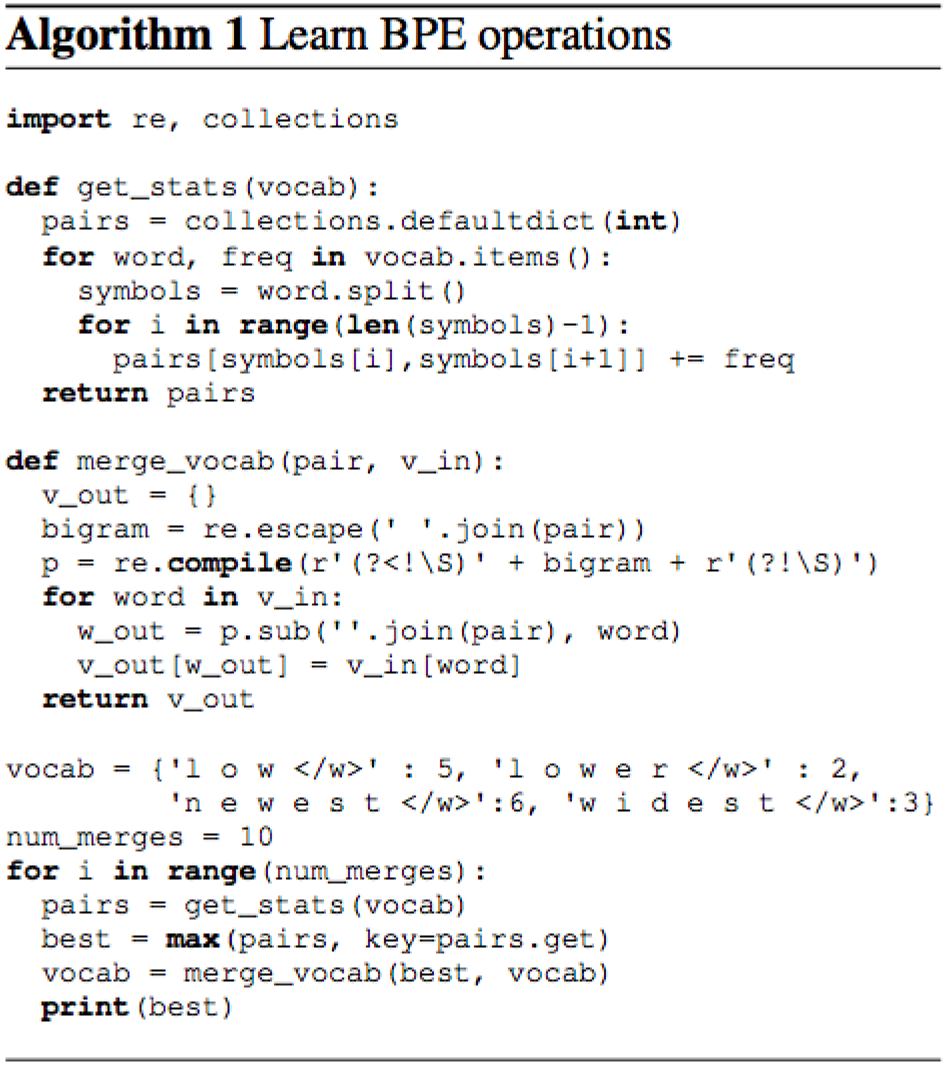
- Word
  - Out-Of-Vocabulary (OOV)
- Character
  - Long sequence ([‘T’, ‘h’, ‘e’, ‘d’, ‘e’, ‘v’, ‘i’, ‘l’, ‘i’, ‘s’, ‘i’, ‘n’, ‘t’, ‘h’, ‘e’, ‘d’, ‘e’, ‘t’, ‘a’, ‘i’, ‘l’, ‘s’])
  - Low performance
- Subword
  - Shorter sequence
  - Higher performance
  - (Almost) Free from OOV

Subword-based Algorithms

- Byte-pair Encoding (BPE)
  - Statistical method (e.g., GPT)
- WordPiece
  - Merge a pair that maximizes the likelihood of the training data once added to the vocab (e.g., BERT, DistillBERT, ELECTRA)
  - E.g., p('ug')/p('u' then 'g')
- Unigram
  - Starts from pretokenized words and the most common substrings then trims
- SentencePiece
  - Solve the problem that all the above methods are using space to separate words before tokenizatio n by dealing with the space as a token (e.g., ALBERT, XLNet, T5)

Byte-pair Encoding

- Byte-pair Encoding (BPE)
  - Statistical method (e.g., GPT)



Algorithm 1 Learn BPE operations

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'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}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

Issue

- Token after space
    - Example
        - Knowledge graph is one of the external resource in language generation task.
    - Substring match problem
        - Example
            - Sulfhydration
            - 학교에