Applied Deep Learning

More on Embeddings

March 31st, 2020 http://adl.miulab.tw

國立臺灣大學
National Taiwan University

2 Handling Out-of-Vocabulary

- One of the main problems of using pre-trained word embeddings is that they are unable to deal with out-of-vocabulary (OOV) words, i.e. words that have not been seen during training.
- Typically, such words are set to the **UNK** token and are assigned the same vector, which is an ineffective choice if the number of OOV words is large.

3

Below Words

Subwords and characters

Subword Embeddings

- separating unseen or rare words into common subwords, potentially address OOV issue
- “AppleCare” = “Apple” + “Care”, “iPhone11” = “iPhone” + “11”

Why Subwords?

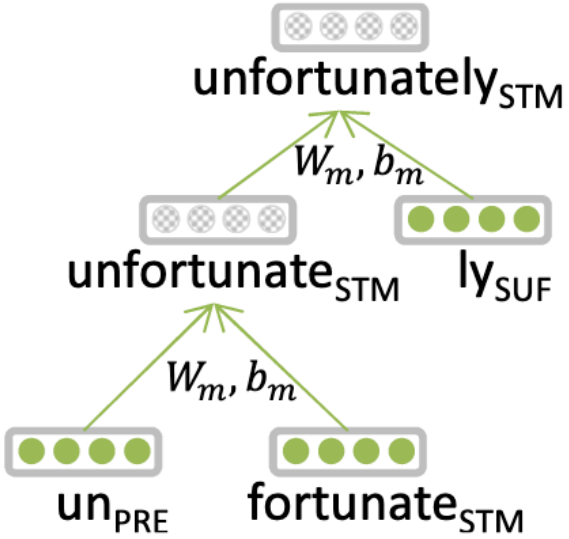
- “台灣大學生喜歡深度學習”
- suboptimal word segmentation system
- ambiguity in word segmentation: “深度學習” or “深度” “學習”
- informal spelling: ”So goooooooood.”, “lolllllllll”

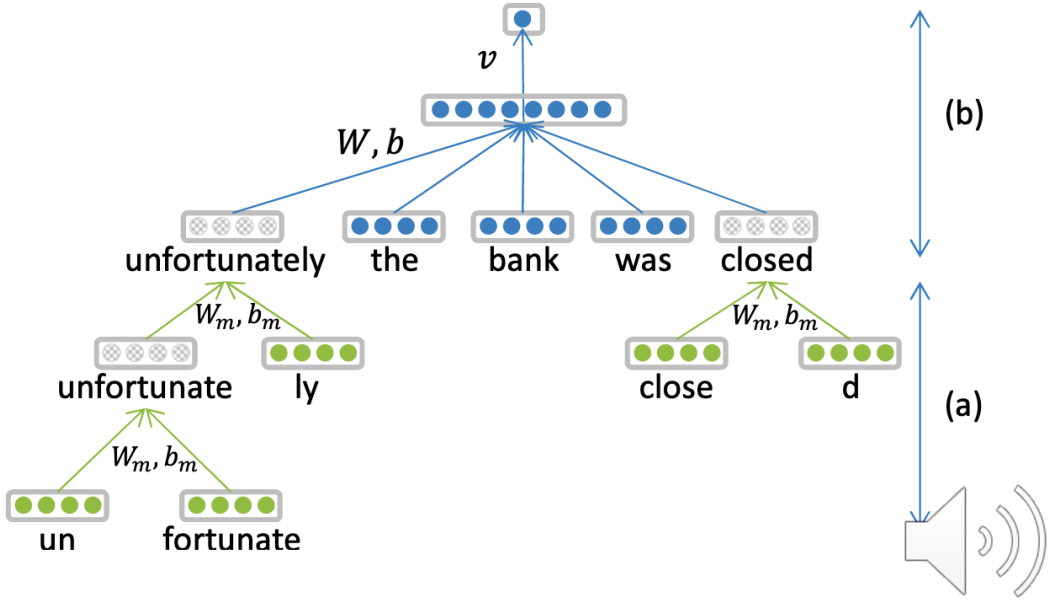
Subword Embeddings

- Possibility of leveraging **morphological** information
- In speech, we have phonemes; in language, we have morphemes.
- Morphemes (語素): smallest semantic units
- **-s**: noun plural, **-ed:** verb simple past tense, **pre-, un-…**

Subword Embeddings

- Morphological Recursive Neural Network





How to Decide Subwords?

- by simple n-gram: Apple = [App, ppl, ple]
- **Byte Pair Encoding:** an algorithm to build the vocabulary

Byte Pair Encoding

- Originally a compression algorithm: most frequent byte pair ↦ a new byte.
- Used as a word segmentation algorithm
- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

Byte Pair Encoding

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

Dictionary

5 l o w
2 l o w e r
6 n e w e s t
3 w i d e s t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d.

Byte Pair Encoding

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram

Dictionary

5 low
2 lower
6 newest
3 widest

Vocabulary

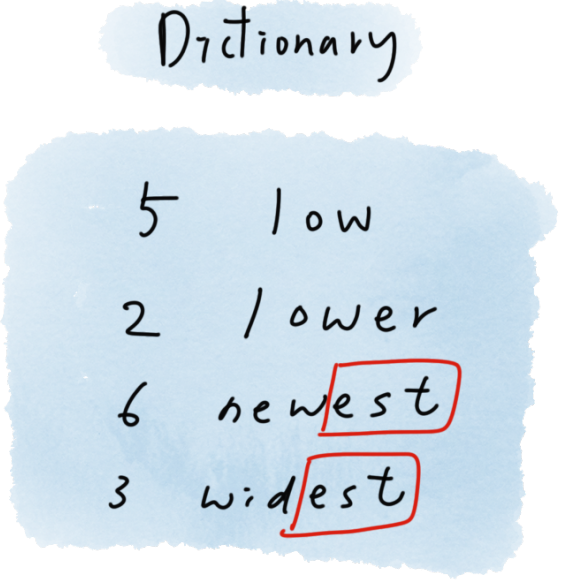
l, o, w, e, r, n, w, s, t, i, d.

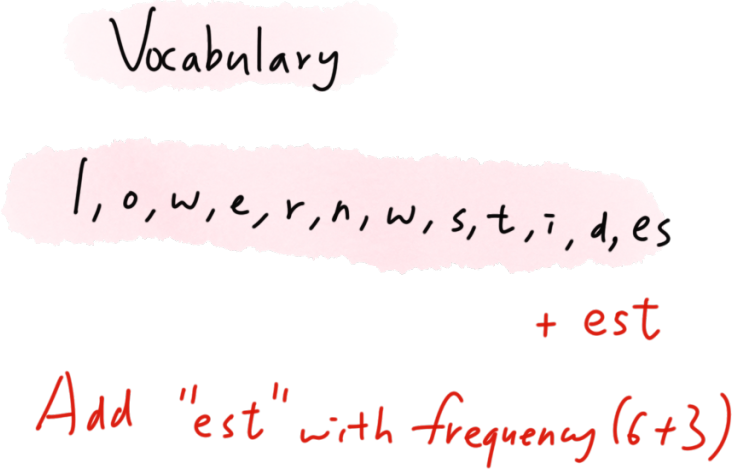
tes

Add "es" with frequency (6+3)

Byte Pair Encoding

- Start with a unigram vocabulary of all (Unicode) characters in data
- Most frequent ngram pairs ↦ a new ngram





Byte Pair Encoding

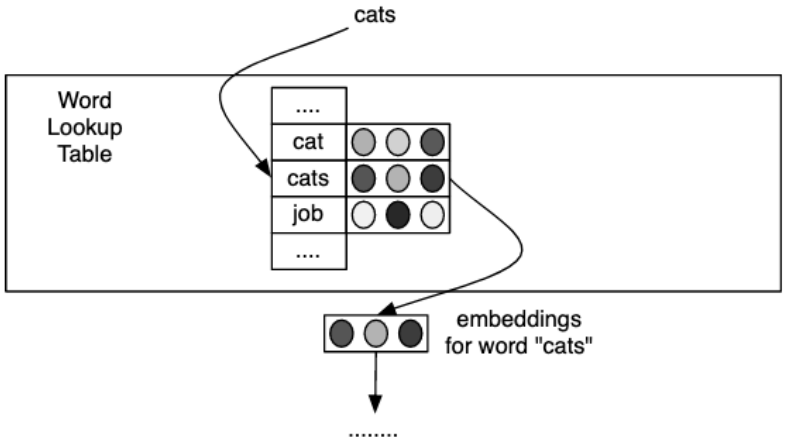
- Have a target vocabulary size and stop when you reach it
- Automatically decides vocab for system

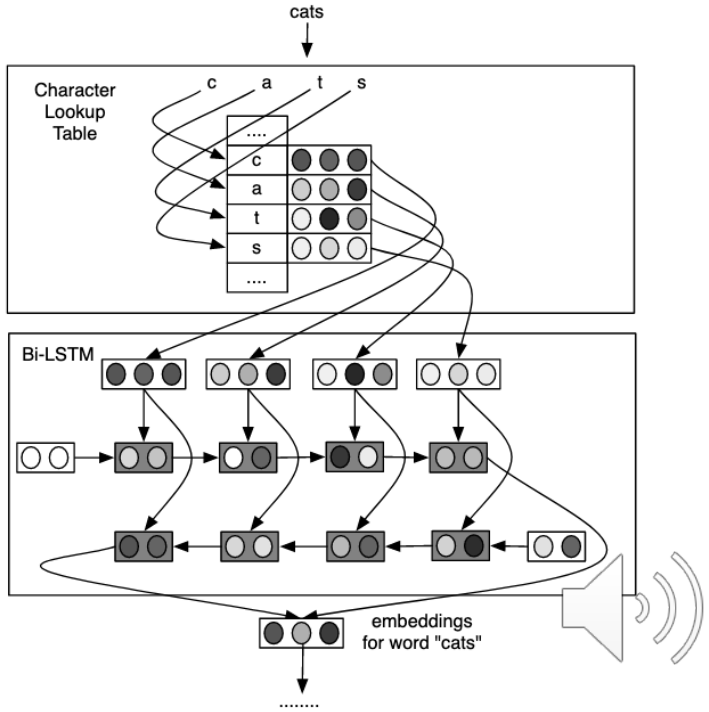
Character-Level Embeddings

- modeling word-level representation by character-level information
- completely solve OOV problem
- dynamically infer representation

Character-Level Embeddings

- compositional character to word (C2W) model





MIMICK

- Optimizing towards pretrained embeddings
- no need to access the originating corpus

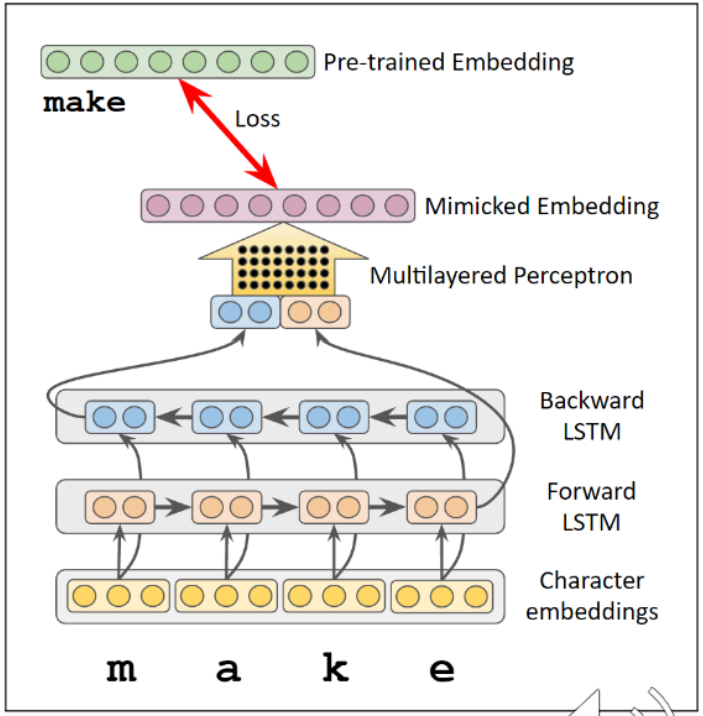


Figure 1: MIMICK model architecture.

FastText

- An extension of the word2vec skip-gram model with character n-grams
- Represent word as char n-grams augmented with boundary symbols and as whole word: Apple = [<Ap, App, ppl, ple, le>, Apple]
- Prefix, suffixes and whole words are special
- supervised objective: text classification

18

Beyond Words

Sentences and documents

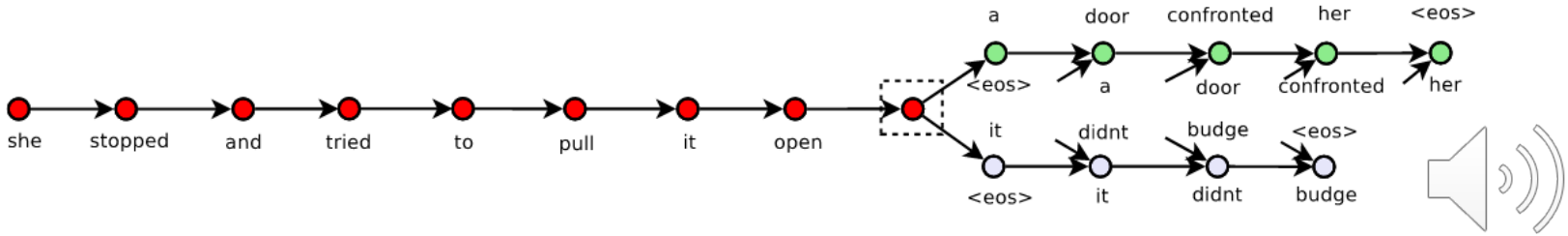
Sentence/Document Embedding

- How to extend to sentence/document-level?
- simply averaging word embeddings, inferring by trained models, … etc.
- training objective?

Skip-Thought

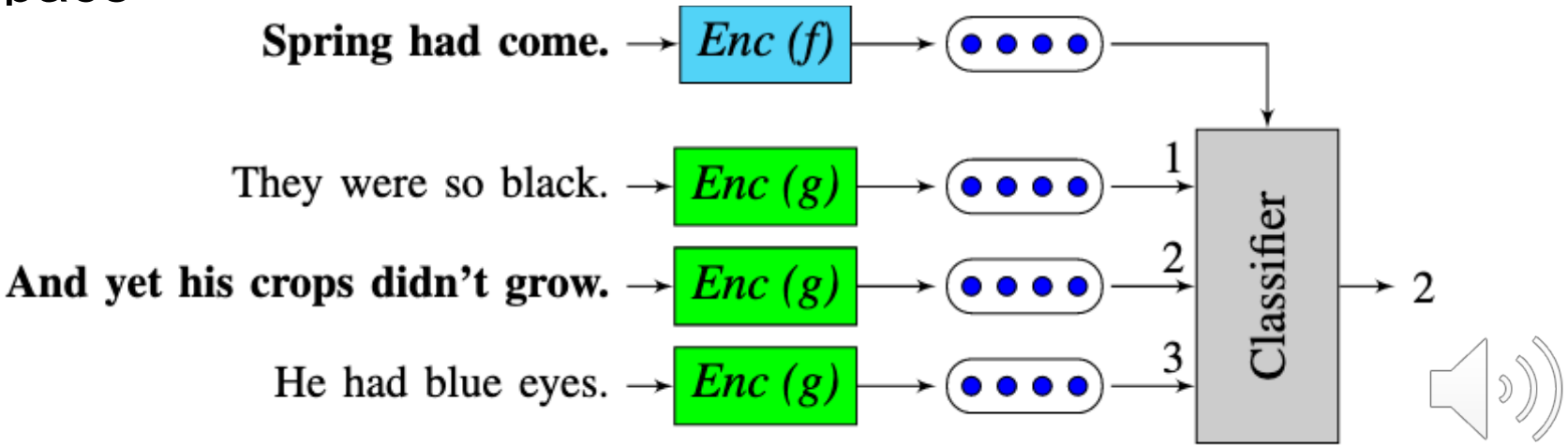
- extend skip-gram concept to sentence-level
- inspired by the distributional hypothesis: sentences that have similar surrounding context are likely to be both semantically and syntactically similar

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i)$$



Quick-Thought

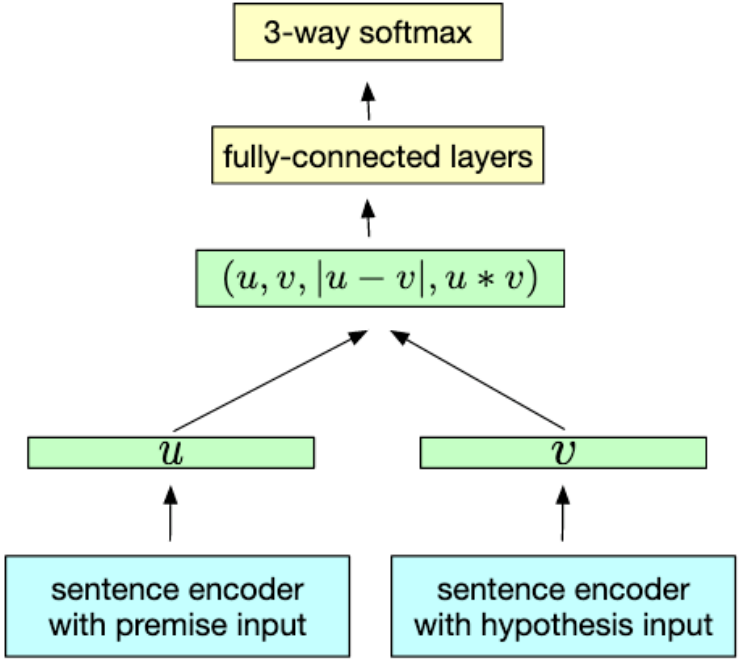
- change the objective to classification problem
- the model can choose to ignore aspects of the sentence that are irrelevant in constructing a semantic embedding space

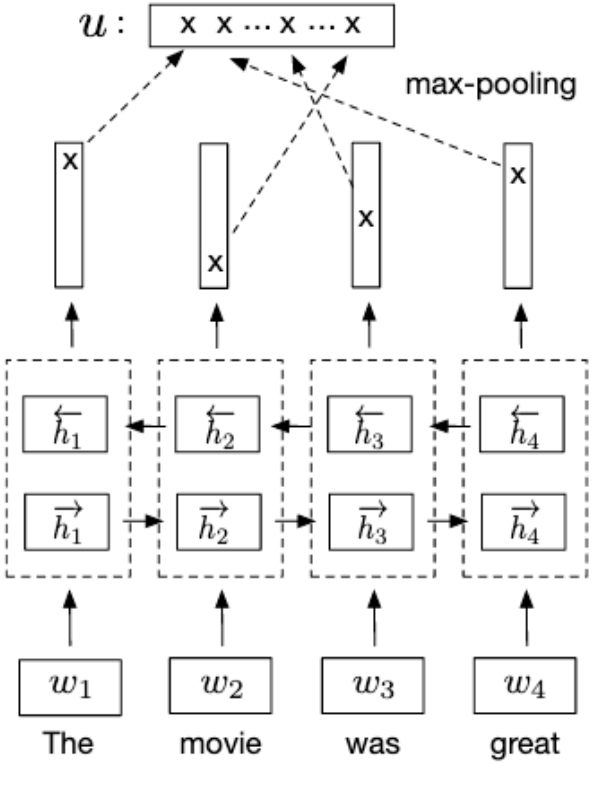


InferSent

- trained on natural language inference (NLI) task
- NLI is the task of determining whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”.

InferSent


3-way softmax
fully-connected layers
$(u, v, |u - v|, u * v)$
u
v
sentence encoder with premise input
sentence encoder with hypothesis input


u :
x x ··· x ··· x
max-pooling
$\overleftarrow{h_1}$ $\overleftarrow{h_2}$ $\overleftarrow{h_3}$ $\overleftarrow{h_4}$
$\overrightarrow{h_1}$ $\overrightarrow{h_2}$ $\overrightarrow{h_3}$ $\overrightarrow{h_4}$
w_1 w_2 w_3 w_4
The movie was great

References

- https://www.aclweb.org/anthology/W13-3512.pdf
- http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture12-subwords.pdf
- http://www.aclweb.org/anthology/D15-1176
- https://arxiv.org/pdf/1508.07909.pdf
- https://arxiv.org/pdf/1707.06961.pdf
- https://github.com/Separius/awesome-sentence-embedding
- https://openreview.net/pdf?id=rJvJXZb0W
- https://arxiv.org/pdf/1607.01759.pdf

References

- https://arxiv.org/pdf/1705.02364.pdf