If Content is King,
Context is God!

More on Embeddings

Mar 26th, 2019

Applied Deep Learning

SHANG-YU SU

HTTP://ADL.MIULAB.TW

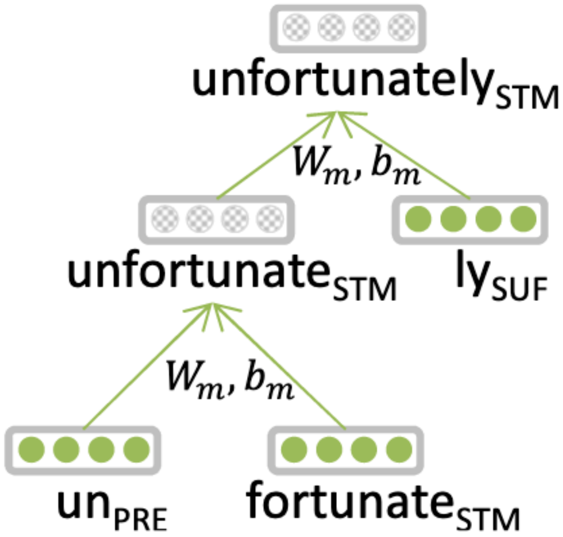
國立臺灣大學
National Taiwan University

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.

Subword-Level Embeddings

- separating unseen or rare words into common subwords, potentially address OOV issue
- “AppleCare” = “Apple” + “Care”, “unfortunately” = “un” + “fortunate” + ” ly”
- Possibility of leveraging morphological information
- Morphological Recursive Neural Network



Why Subwords?

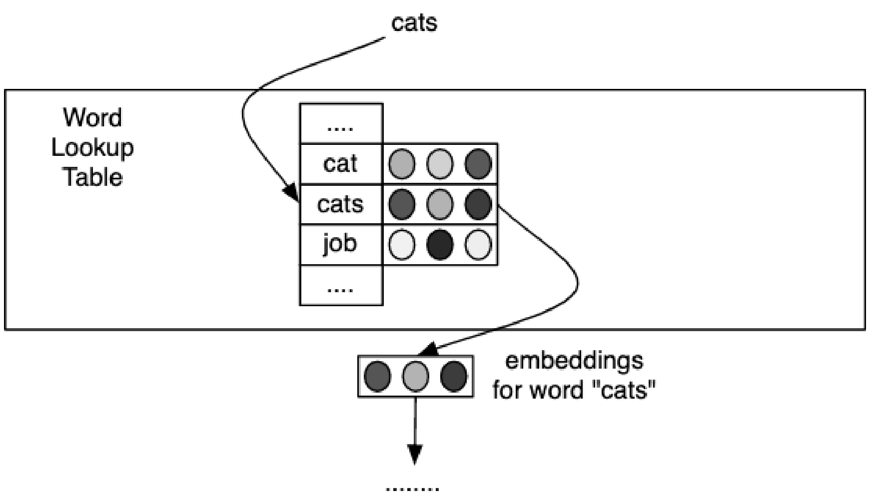
- “台灣大學生喜歡深度學習”
- suboptimal word segmentation system
- ambiguity in word segmentation: “深度學習” or “深度” “學習”
- informal spelling: ”So gooooooood.”, “lollllllll”

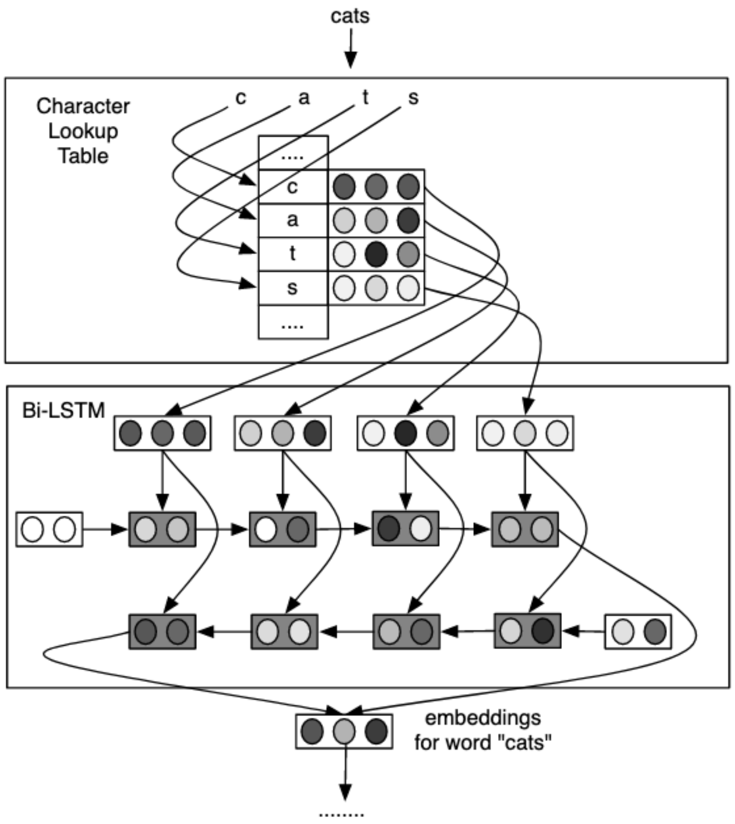
How to Decide Subwords?

- by n-gram: Apple = [App, ppl, ple]
- Automatically decides vocab for system: **Byte Pair Encoding**
- Most frequent n-gram pairs $\mapsto$ a new n-gram

Character-Level

- modeling word-level representation by character-level information
- completely solve OOV problem
- end-to-end training
- dynamically infer representation by RNN





Character-Level

- modeling word-level representation by character-level information
- completely address OOV
- MIMICK Word Embeddings
- no need to access the originating corpus

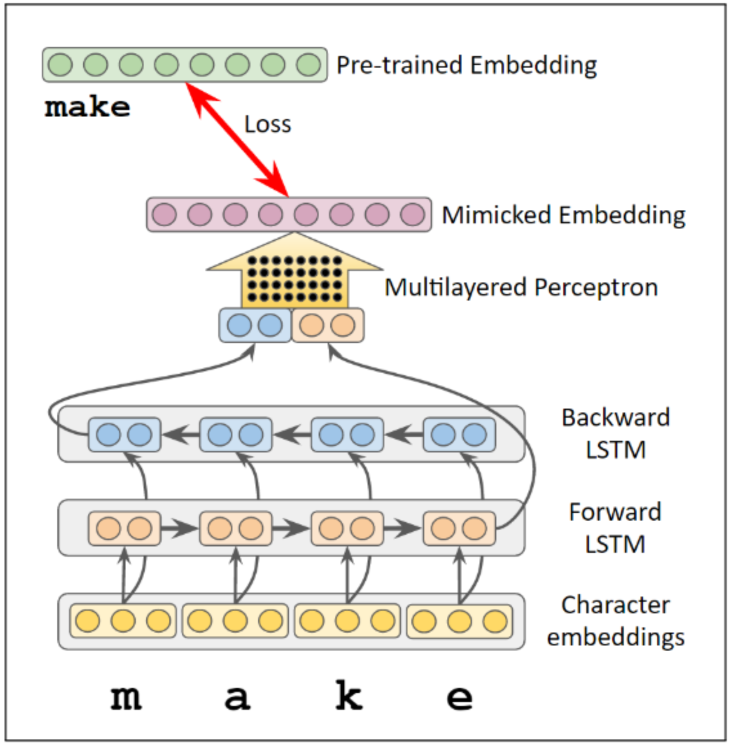
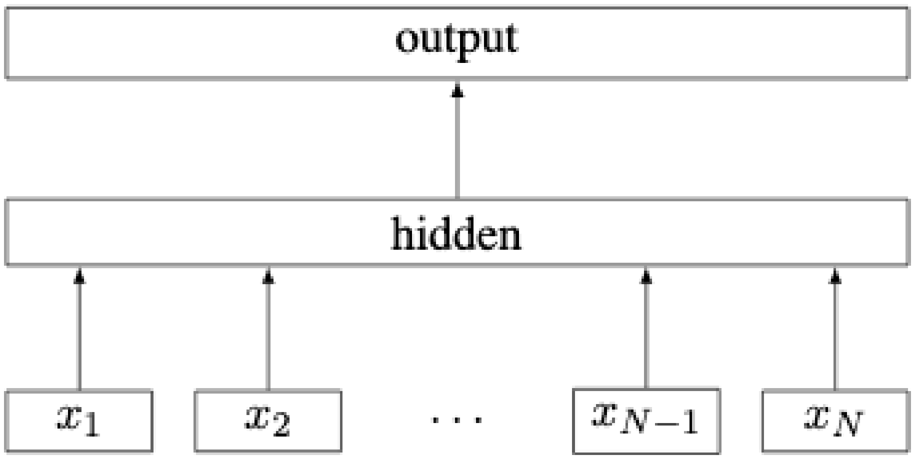


Figure 1: MIMICK model architecture.

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**



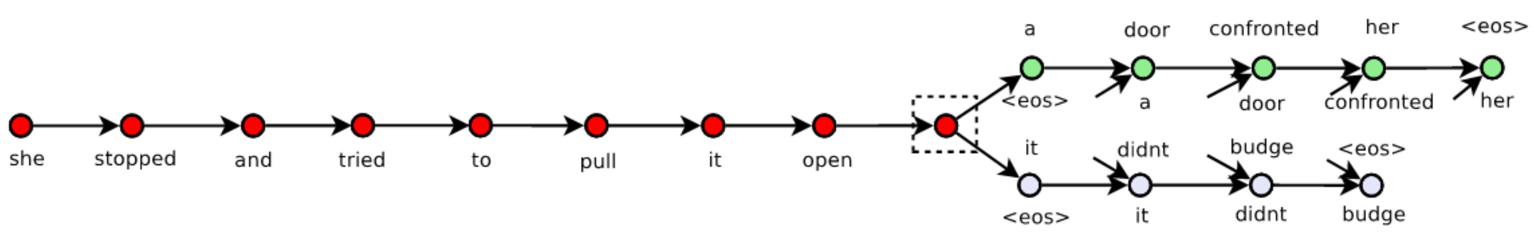
Sentence/Document Embedding

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

Skip-Thought

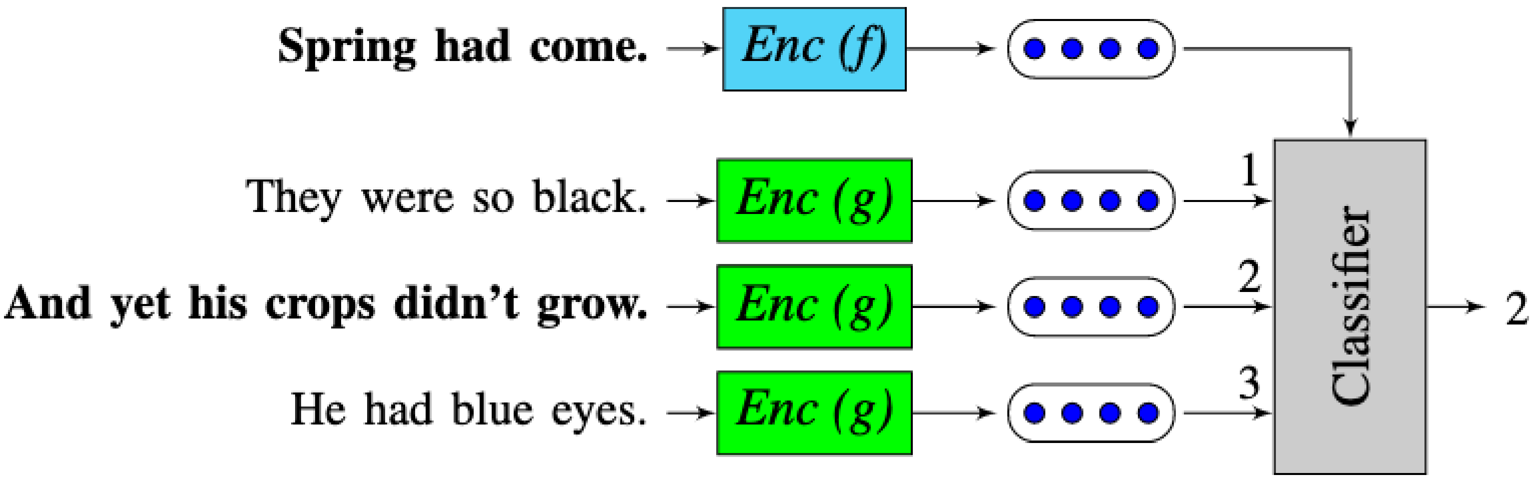
- extend skip-gram concept to sentence-level
- inspired by the distributional hypothesis: sentences that have similar surrounding context are likely to be both semantically and syntactically similar

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



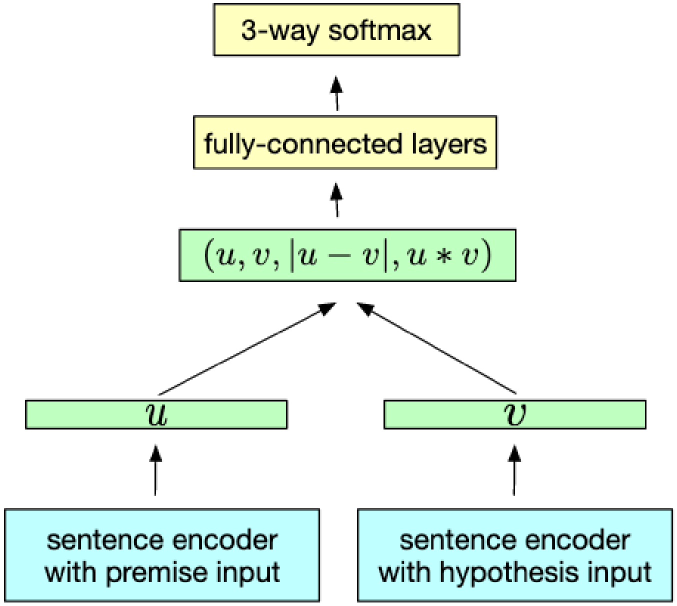
Quick-Thought

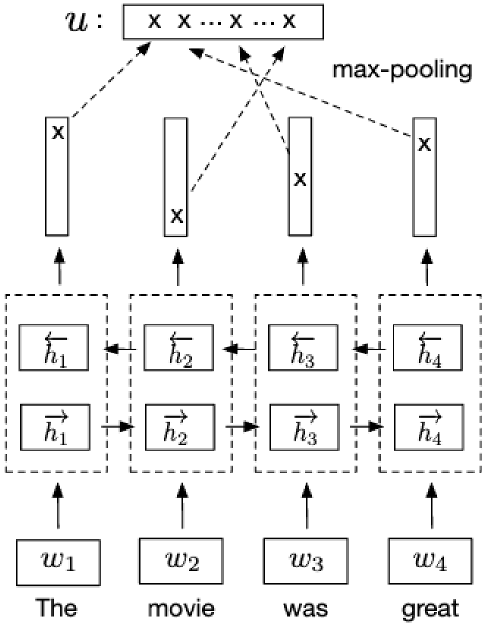
- change the objective to classification problem
- the model can choose to ignore aspects of the sentence that are irrelevant in constructing a semantic embedding space



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

- so what is the best objective/task to learn generalized representation?
- should we train the model? 😅

BiLSTM-Max (untrained)†	77.5	**81.3**	89.6	**88.7**	80.7	**85.8**	73.2/81.6	**0.860**	83.4	.39/.48
Unsupervised representation training (ordered sentences)										
FastSent	70.8	78.4	88.7	80.6	-	76.8	72.2/80.3	-	-	**.63/.64**
FastSent+AE	71.8	76.7	88.8	81.5	-	80.4	71.2/79.1	-	-	.62/.62
SkipThought	76.5	80.1	93.6	87.1	82.0	**92.2**	**73.0/82.0**	**0.858**	82.3	.29/.35
SkipThought-LN	**79.4**	**83.1**	**93.7**	**89.3**	82.9	88.4	-	**0.858**	79.5	.44/.45
Supervised representation training										
CaptionRep (bow)	61.9	69.3	77.4	70.8	-	72.2	73.6/81.9	-	-	.46/.42
DictRep (bow)	76.7	78.7	90.7	87.2	-	81.0	68.4/76.8	-	-	**.67/.70**
NMT En-to-Fr	64.7	70.1	84.9	81.5	-	82.8	69.1/77.1	-		.43/.42
Paragram-phrase	-	-	-	-	79.7	-	-	0.849	83.1	**.71**/ -
BiLSTM-Max (on SST)†	(*)	83.7	90.2	89.5	(*)	86.0	72.7/80.9	0.863	83.1	.55/.54
BiLSTM-Max (on SNLI)†	79.9	84.6	92.1	**89.8**	83.3	**88.7**	75.1/82.3	**0.885**	**86.3**	.68/.65
BiLSTM-Max (on AllNLI)†	**81.1**	**86.3**	**92.4**	**90.2**	**84.6**	88.2	**76.2/83.1**	**0.884**	**86.3**	**.70/.67**

Smooth Inverse Frequency (SIF)

- key ideas: smooth inverse frequency weighting (W) and common component removal (R)
- **no need to train**

Algorithm 1 Sentence Embedding

Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter a and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.

Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$

```
for all sentence s in S do
    v_s ← (1/|s|) Σ_{w∈s} (a / (a + p(w))) v_w
end for
Form a matrix X whose columns are {v_s : s ∈ S}, and let u be its first singular vector
for all sentence s in S do
    v_s ← v_s − u u^T v_s
end for
```

References

- http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture12-subwords.pdf
- http://www.aclweb.org/anthology/W13-3512
- 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=SyK00v5xx
- https://openreview.net/pdf?id=rJvJXZb0W
- https://arxiv.org/pdf/1607.01759.pdf
- https://arxiv.org/pdf/1705.02364.pdf