Document Classification by Inversion of Distributed Language Representations

Matt Taddy, Chicago Booth

Distributed Language Representation

$\mathcal{V}$ contains an embedding in $\mathbb{R}^K$ for every vocabulary word.

In a contextual language model, $\mathcal{V}$ is trained to maximize the likelihoods for each single words and its neighbors.

e.g., The skip-gram objective for word t in sentence s is

$$\max \sum_{j \neq t,\ j=t-b}^{t+b} \log \mathrm{p}_{\mathcal{V}}(w_{sj} \mid w_{st})$$

where b is the skip-gram window (truncate at ends of sentences).

Neural network language models

Local context probabilities are functions of the word embeddings.

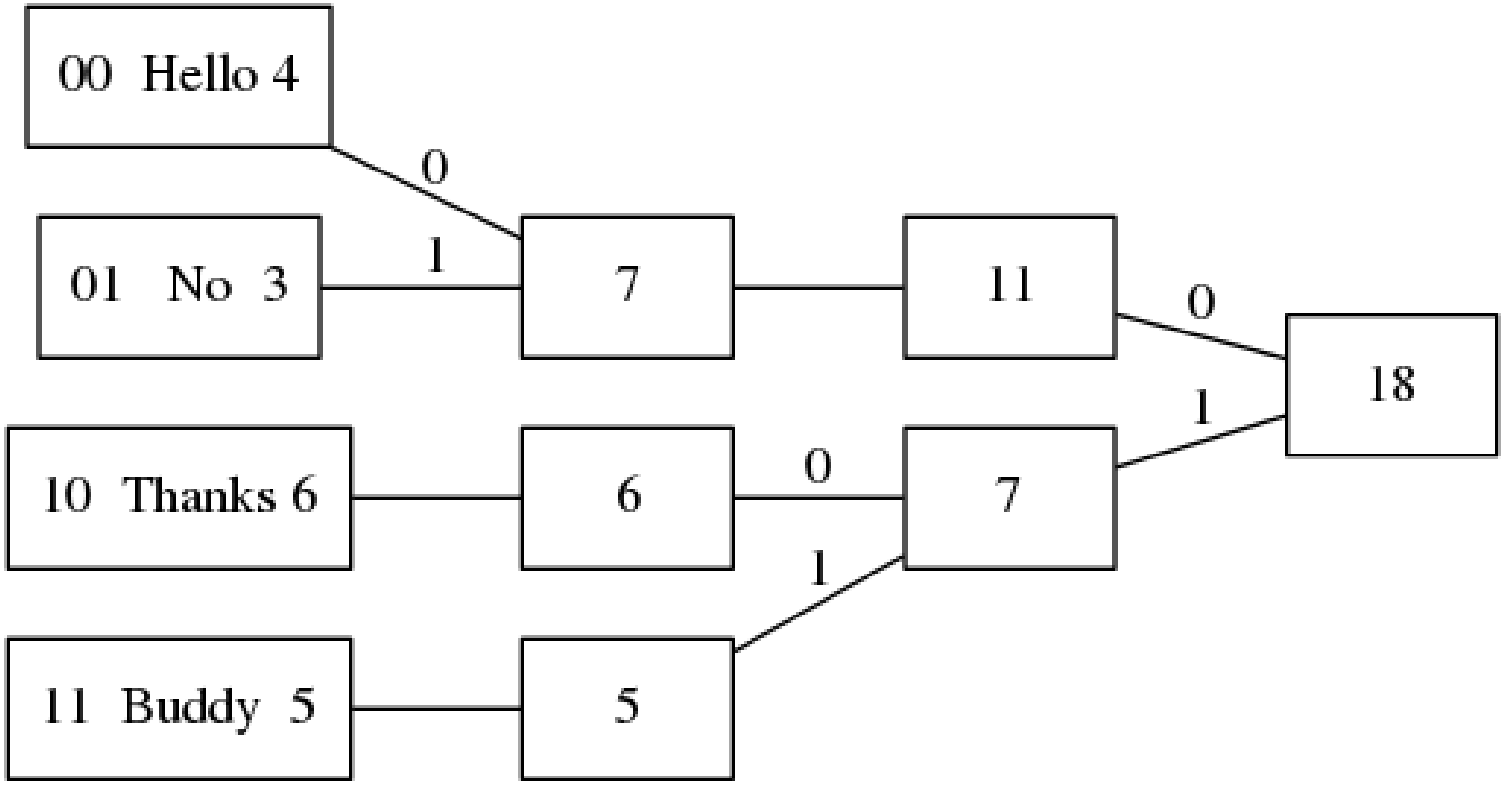
e.g., in Word2Vec

$$\mathrm{p}_{\mathcal{V}}(w|w_t) = \prod_{j=1}^{L(w)-1} \sigma\left(\mathrm{ch}\left[\eta(w, j+1)\right] \mathbf{u}_{\eta(w,j)}^{\top} \mathbf{v}_{w_t}\right)$$

where $\eta(w, i)$ is the i^{th} node in the length $L(w)$ Huffman tree path for w and $\mathrm{ch}(\eta) \in \{-1, +1\}$ for whether η is a left or right child.

'Output' embedding $\mathbf{v}_{w_t}$ is usually the main object of interest.

Example Huffman encoding of a 4 word vocabulary



From left to right the two nodes with lowest count are combined into a parent. Encodings are read off of the splits from right to left.