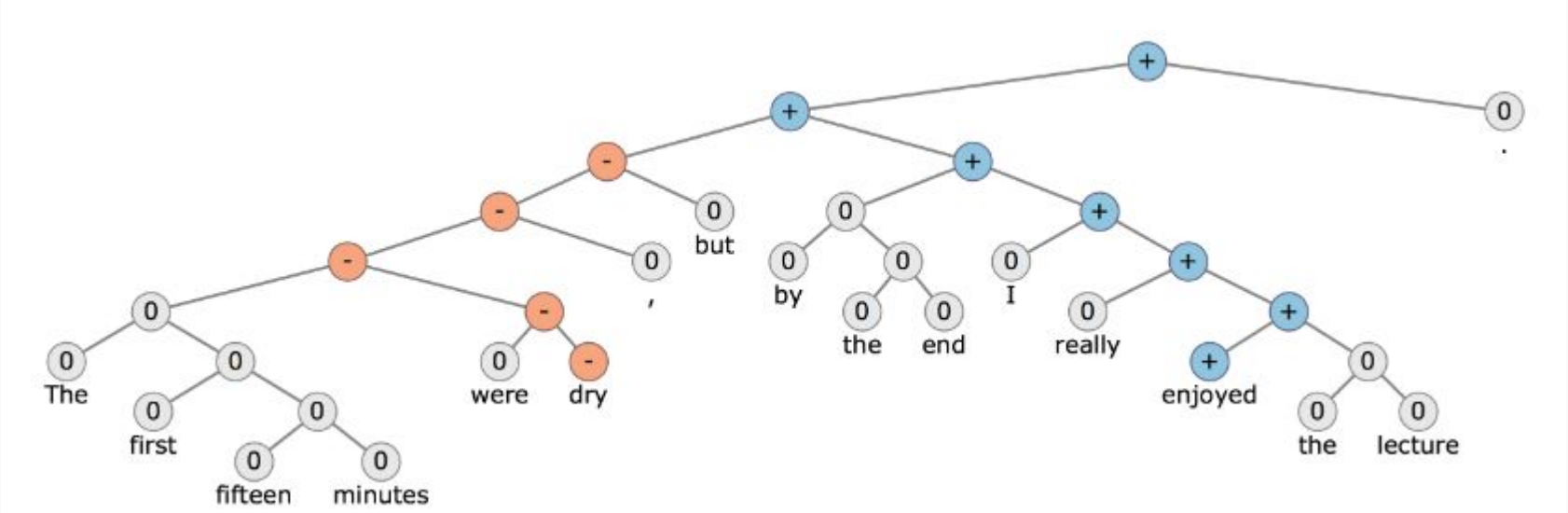
Bag of Tricks for Efficient Text Classification

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov
Facebook AI Research

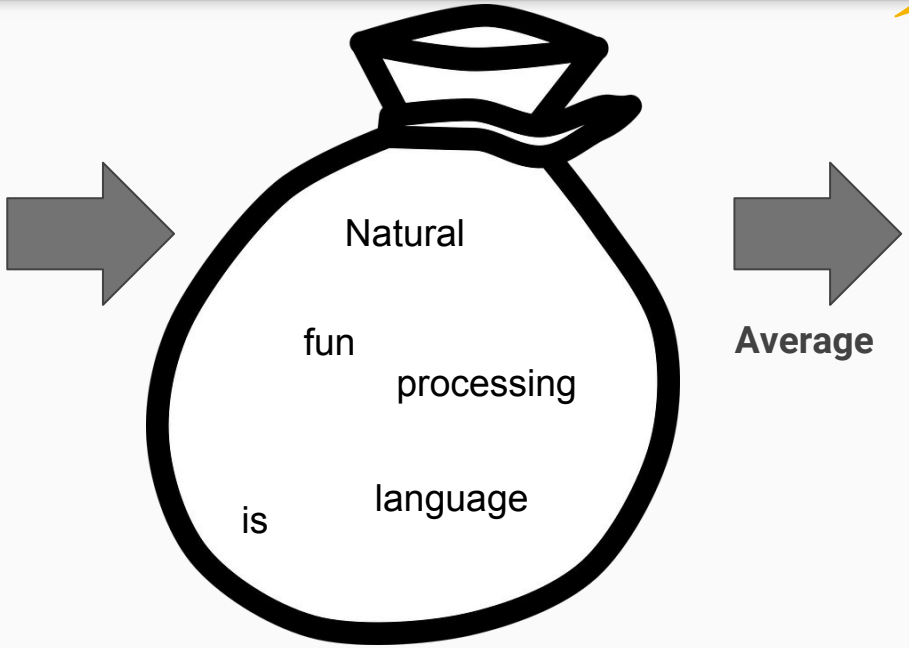
Text classification



Bag of Words (or n-grams)

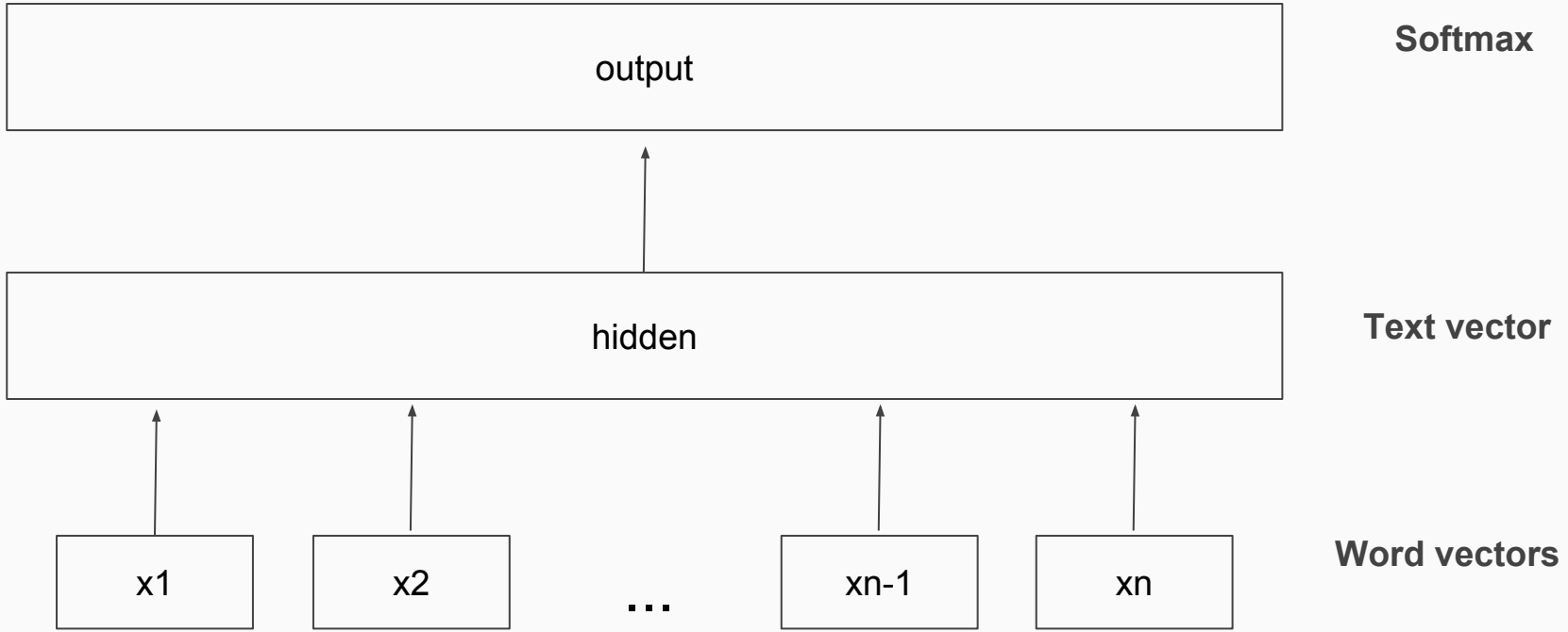
low-dimensional!

Natural language processing is fun.

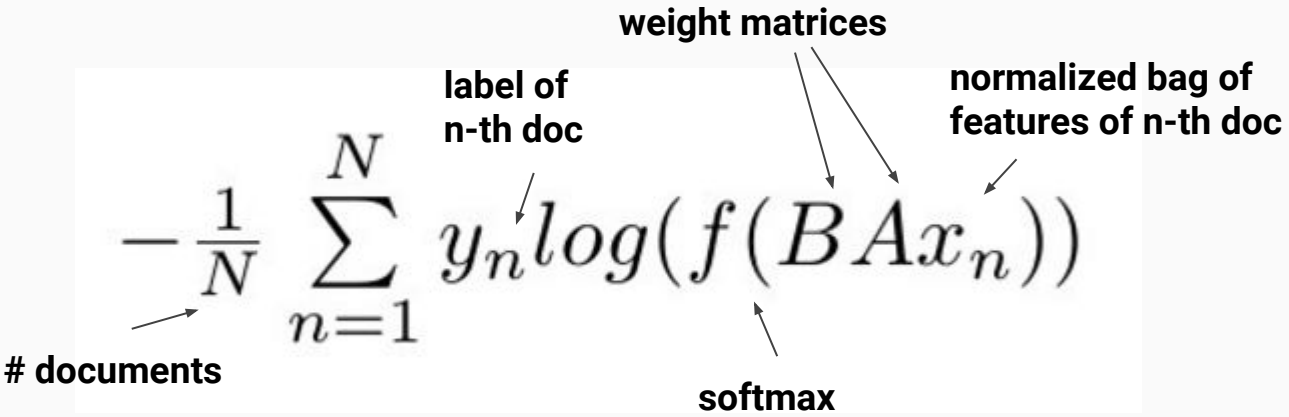


$$\begin{pmatrix} -0.132 \\ 1.129 \\ 0.827 \\ 0.110 \\ -0.527 \\ 0.156 \\ 0.349 \\ -0.286 \end{pmatrix}$$

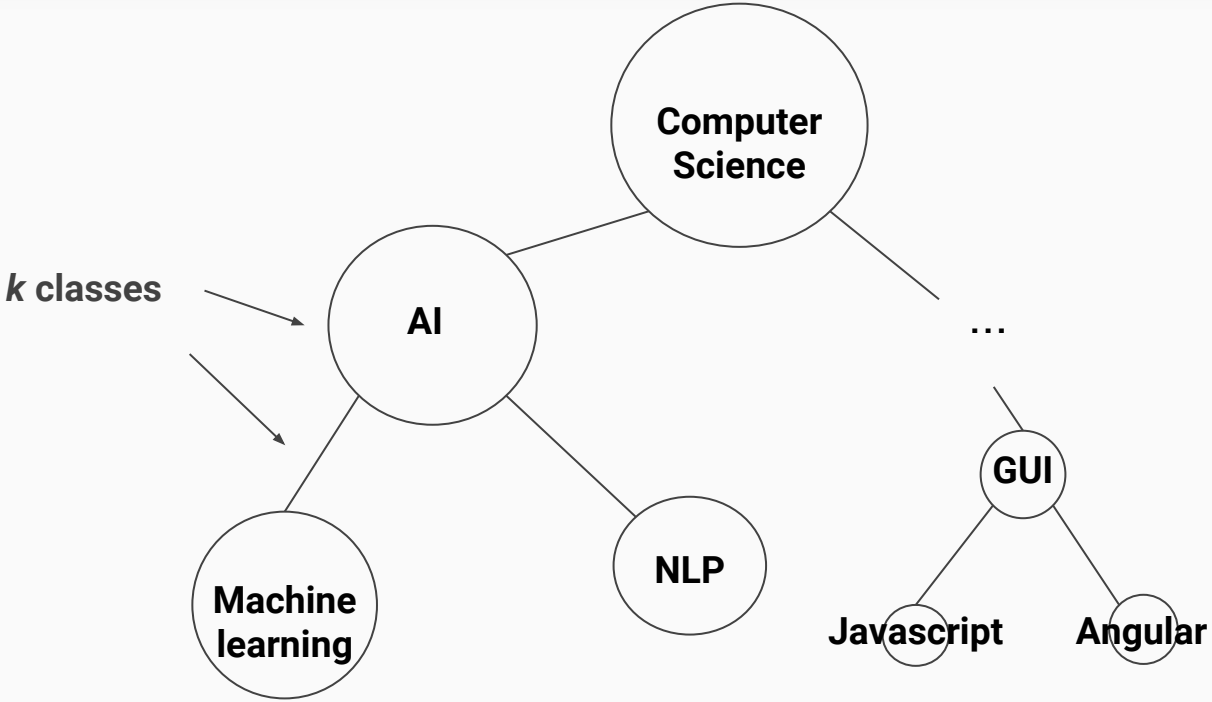
Simple linear model



Learning


weight matrices
label of
n-th doc
normalized bag of
features of n-th doc
$-\frac{1}{N}\sum_{n=1}^{N} y_n log(f(BAx_n))$
documents
softmax

Hierarchical softmax



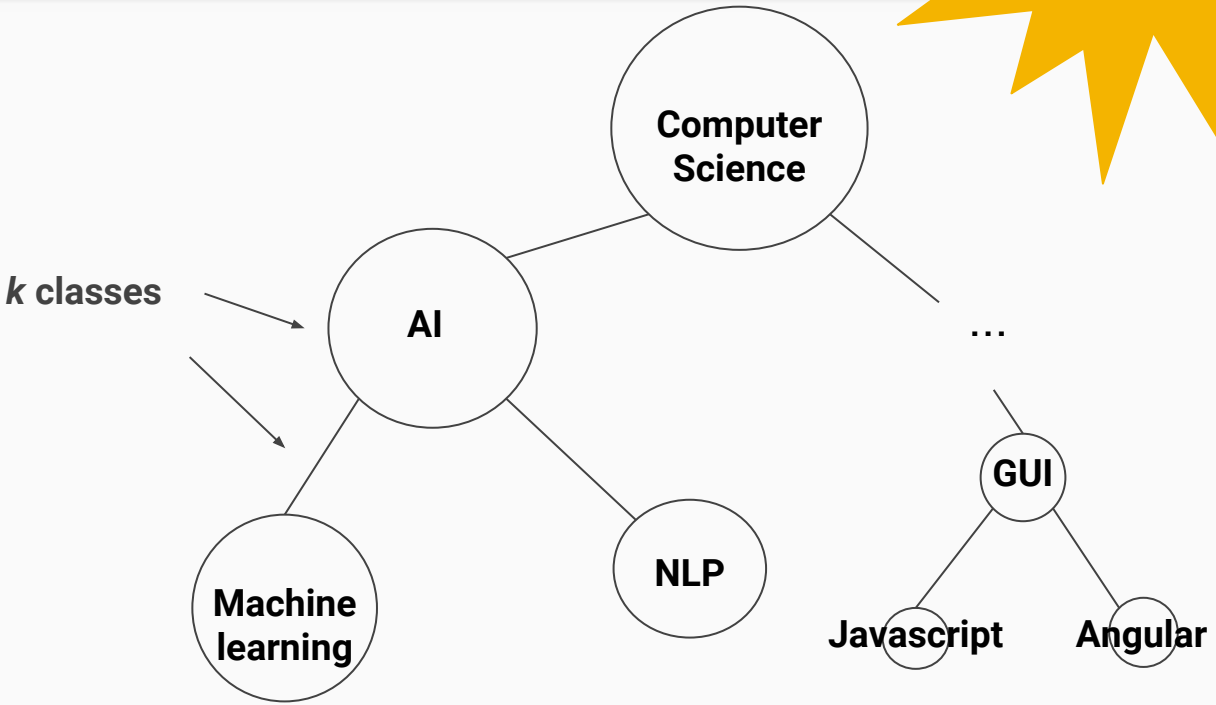
$$P(n_{l+1}) = \prod_{i=1}^{l} P(n_i).$$

Probability of a node is always lower than one of its parent

Hierarchical softmax

O(hlog(k)) vs **O(kh)** training time!

(Text representation dimension *h*)



$$P(n_{l+1}) = \prod_{i=1}^{l} P(n_i).$$

Probability of a node is always lower than one of its parent

Results

As good as NN!

	Yahoo		Amazon full		Amazon polarity	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
char-CNN	71.2	1 day	59.5	5 days	94.5	5 days
VDCNN	73.4	2h	63	7h	95.7	7h
fastText	72.3	5s	60.2	9s	94.6	10s

Fast!

Summary

- fastText is often on par with deep learning classifiers
- fastText takes seconds, instead of days
- Can learn vector representations of words in different languages (with performance better than word2vec!)

Thanks!