Sentiment Analysis and Basic Feature Extraction

the movie was great! would watch again!

the film was awful; I'll never watch again!

① text $\bar{x} \Rightarrow f(\bar{x})$ feature extraction

② $\{f(\bar{x}^{(i)}), y^{(i)}\}_{i=1}^{D}$ dataset of D labeled exs,

$\Rightarrow$ train classifier

Feature extraction

the movie was great

Bag-of-words: Assume 10,000 words in vocabulary

$$\left[\begin{array}{ccccccccccc} 1 & 0 & 0 & 0 & & 1 & & 1 & & 1 & \\ \text{the} & \text{a} & \text{of} & \text{at} & \ldots & \text{movie} & \ldots & \text{was} & \ldots & \text{great} & \ldots \end{array}\right]$$

4 1s
9996 0s

Counts (how many "the" are present)

presence/absence (0/1)

Bag-of-ngrams

n-gram: sequence of n consecutive words

2-grams: <u>the movie</u>, <u>movie was</u>, <u>was great</u>

tf-idf
↳ tf × idf

tf: count of the term

idf: inverse document frequency $\log \frac{\overset{\text{total \# docs}}{N}}{\underset{\text{\# docs with } w}{\{D : w \in D\}}}$

Preprocessing

[... great ..... great! ...]

① Tokenization

was great!

was great

→ was␣great␣!

wasn't → was␣n't

② [Sometimes] Stopword removal (the, of, a, ...)

③ [Sometimes] Casing (lowercasing, truecasing)

④ Handling unknown words   Durrett ⇒ UNK

⑤ Indexing: map each {word, n-gram} into $\mathbb{N}$

use a map