Preprocesamiento de texto

Aprendizaje Automático

Docente: Juan David Martínez Vargas
jdmartinev@eafit.edu.co

Preprocesamiento de texto

Raw text data

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

↓

Preprocessed text data

(e.g., strip HTML)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

↓

Feature vector

□□□□□□ ... □□□□□□

↓

Machine learning model

Preprocesamiento de texto

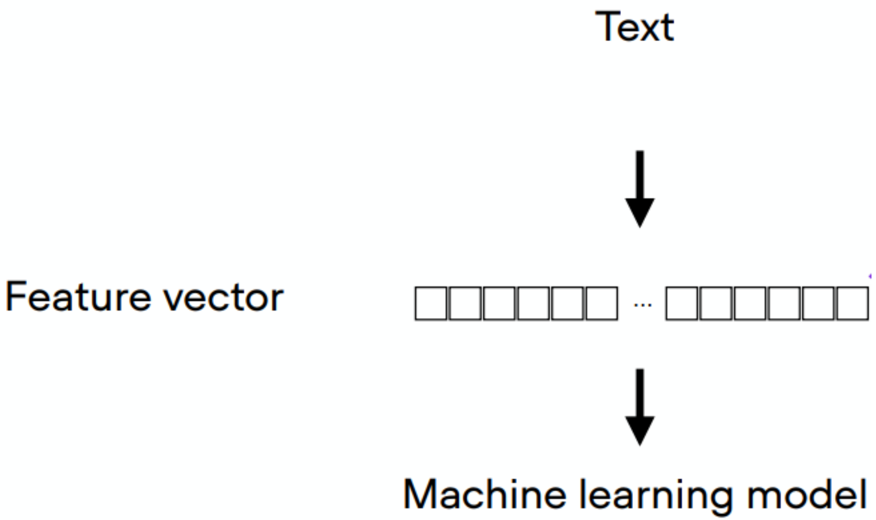
Machine learning model

Classic methods for tabular data:
- Logistic regression
- Multilayer perceptron

Sequence models:
- 1D convolutional network
- Recurrent neural network

Transformers

Preprocesamiento de texto



1. Bag of words
2. Embeddings

Tokenización

Divide el texto/documento en partes pequeñas con espacios en blanco y puntuaciones

Sentence	Tokens
“I don’t like eggs.”	“I”, “do”, “n’t”, “like”, “eggs”, “.”

Remoción de stop words

Stop words: Palabras que aparecen frecuentemente en textos pero no contribuyen mucho al significado de las oraciones

- Stop words comunes (eng): “a”, “the”, “so”, “is”, “it”, “at”, “in”, “this”, “there”, “that”, “my”

Original sentence	Without stop words
“There is a tree near the house”	“tree near house”

Bag of words

Text	Label
The ghost pepper is so spicy, it is hauntingly hot	1
I tried to hug the sun today, but it was too hot to handle	1
I cannot handle spicy food	0

Bag of words

Text	Label
The ghost pepper is so spicy, it is hauntingly hot	1
I tried to hug the sun today, but it was too hot to handle	1
I cannot handle spicy food	0

Vocabulary

```
but
cannot
food
ghost
handle
hauntingly
hot
hug
i
is
it
pepper
so
spicy
sun
the
to
today
too
tried
was
```

Bag of words

- El número de palabras determina el número de características
- El número de veces que aparece cada palabra en cada ejemplo determina el valor de cada característica

Text
The ghost pepper is so spicy, it is hauntingly hot
I tried to hug the sun today, but it was too hot to handle
I cannot handle spicy food

but	cannot	food	ghost	handle	hauntingly	hot	hug	i	is	it	pepper	so	spicy	sun	the	to	today	too	tried	was
0	0	0	1	0	1	1	0	0	1	1	1	1	1	0	1	0	0	0	0	0
1	0	0	0	1	0	1	1	1	0	1	0	0	0	1	1	2	1	1	1	1
0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

BoW vectors

Bag of words

but	cannot	food	ghost	handle	hauntingly	hot	hug	i	is	it	pepper	so	spicy	sun	the	to	today	too	tried	was
0	0	0	1	0	1	1	0	0	1	1	1	1	1	0	1	0	0	0	0	0
1	0	0	0	1	0	1	1	1	0	1	0	0	0	1	1	2	1	1	1	1
0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

Feature vectors

Machine
learning model

Modificaciones de Bag of Words

- N - gramas

Text	Label
The ghost pepper is so spicy, it is hauntingly hot	1
I tried to hug the sun today, but it was too hot to handle	1
I cannot handle spicy food	0

"The"
"ghost"
"pepper"
"is"
"so"
"spicy"
...

1-gram

"The ghost"
"ghost pepper"
"pepper is"
"is so"
"so spicy"
...

2-gram
(bigram)

Modificaciones de Bag of Words

tf-idf: Term frequency – inverse document frequency

Qué tan a menudo aparece una palabra, **ponderado** por el número de documentos en el que la palabra aparece:

- Frecuencia alta de aparición de la palabra: palabra importante
- Frecuencia alta de aparición en documentos: no tan informativa

Embeddings

One-hot encoded ("sparse") representation of "S U N N Y"

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Embedded ("dense") representation of "S U N N Y"

```
[[0.9816, 0.7363, 0.5899],
 [0.2605, 0.3766, 0.3502],
 [0.7382, 0.9807, 0.4762],
 [0.6231, 0.8825, 0.8836]]
```

Embedding layer

```
  [[0.6912, 0.8765, 0.4939],
    [0.6342, 0.7481, 0.7717],
    [0.8395, 0.2128, 0.3696],
    [0.4900, 0.1509, 0.0689],
    [0.2587, 0.9171, 0.8670],
    [0.7213, 0.9922, 0.5701],
    [0.7598, 0.5231, 0.3666],
    [0.5150, 0.5216, 0.9682],
    [0.2248, 0.0261, 0.4427],
    [0.1818, 0.6863, 0.8713],
    [0.4192, 0.1566, 0.9004],
    [0.8102, 0.5741, 0.4241],
    [0.1116, 0.0466, 0.2786],
S   [0.9816, 0.7363, 0.5899],
    [0.9224, 0.3672, 0.6972],
    [0.1207, 0.3372, 0.2128],
    [0.0660, 0.1524, 0.8440],
    [0.2162, 0.5640, 0.0988],
U   [0.2605, 0.3766, 0.3502],
    [0.2334, 0.4757, 0.7581],
N   [0.7382, 0.9807, 0.4762],
    [0.2369, 0.8102, 0.8798],
    [0.6932, 0.2671, 0.8018],
    [0.9593, 0.5302, 0.4290],
Y   [0.6231, 0.8825, 0.8836],
    [0.4623, 0.8503, 0.7279]]
```

HuggingFace (pytorch) Embedding layer

Embedding layers: Forma eficiente de multiplicación de matrices cuando se trabaja con vectores codificados de forma one-hot

```
import torch

torch.manual_seed(123);
```

```
idx = torch.tensor([2, 3, 1]) # 3 training examples

num_idx = max(idx)+1
out_dim = 5
```

Input dimension of a one-hot encoded vector is the number of indices (the highest index + 1)

Suppose we want embeddings of size 5

HuggingFace (pytorch) Embedding layer

```
import torch

torch.manual_seed(123);
```

```
idx = torch.tensor([2, 3, 1]) # 3 training examples

num_idx = max(idx)+1
out_dim = 5

embedding = torch.nn.Embedding(num_idx, out_dim)
embedding(idx)
```

```
tensor([[ 0.6957, -1.8061, -1.1589,  0.3255, -0.6315],
        [-2.8400, -0.7849, -1.4096, -0.4076,  0.7953],
        [ 1.3010,  1.2753, -0.2010, -0.1606, -0.4015]],
       grad_fn=<EmbeddingBackward0>)
```

Each training example has 5 feature values