Natural Language Processing

Data Science, Spring 2021

Before we start...

Before we start...

1. Our mod of the day.

Before we start...

1. Our mod of the day.
2. Project 2

Before we start...

1. Our mod of the day.
2. Project 2
3. An equation you were promised

Our moderator

Our moderator

1. Anubhav!

Project 2

Project 2

1. Spend some time on your PDF generation!

An equation you were promised:

$$(1 + \frac{\lambda}{N})^{-2}$$

An equation you were promised:

Let's say you had a variable where half the data was missing ($\lambda = 0.5$) and you used $N = 5$ for the number of generated data sets:

$$(1 + \frac{0.5}{5})^{-2} = 1.049$$

An equation you were promised:

How much better would it be if you used an ‘infinite’ number of generated data sets?

$$(1 + \frac{0.5}{\infty})^{-2} = 1$$

Part I: Text Classification

We often want to classify a text we've been given

Why?

We often want to classify a text we've been given

Why?

1. Determine if something is spam

We often want to classify a text we've been given

Why?

1. Determine if something is spam
2. Sentiment

We often want to classify a text we've been given

Why?

1. Determine if something is spam
2. Sentiment
3. Authorship

We often want to classify a text we've been given

Why?

1. Determine if something is spam
2. Sentiment
3. Authorship
4. Time period of authorship

We often want to classify a text we've been given

Why?

1. Determine if something is spam
2. Sentiment
3. Authorship
4. Time period of authorship
5. What else?

What do we need?

What do we need?

1. A set of classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$

What do we need?

1. A set of classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$
2. Some document $w \in Doc$

What do we need?

1. A set of classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$
2. Some document $w \in Doc$
3. Classification is a function: $classify : Doc \to Y$

Classification

There are many ways to implement such a function:

Classification

There are many ways to implement such a function:

1. Rule-based approach (blacklists, keywords, etc.)

Classification

There are many ways to implement such a function:

1. Rule-based approach (blacklists, keywords, etc.)
2. Supervised learning

Supervised Learning

Supervised Learning

1. Input:

Supervised Learning

1. Input:
    1.1 Document $w \in Doc$

Supervised Learning

1. Input:
   1.1 Document $w \in Doc$
   1.2 Classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$

Supervised Learning

1. Input:
    1.1 Document $w \in Doc$
    1.2 Classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$
    1.3 Training set $T = \{(w_1, y_1), (w_2, y_2), (w_3, y_3) \ldots, (w_n, y_n)\}$

Supervised Learning

1. Input:
   1.1 Document $w \in Doc$
   1.2 Classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$
   1.3 Training set $T = \{(w_1, y_1), (w_2, y_2), (w_3, y_3) \ldots, (w_n, y_n)\}$
2. Output

Supervised Learning

1. Input:
    1.1 Document $w \in Doc$
    1.2 Classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$
    1.3 Training set $T = \{(w_1, y_1), (w_2, y_2), (w_3, y_3) \ldots, (w_n, y_n)\}$
2. Output
    2.1 $classify : Doc \to Y$

Supervised Learning

1. Input:
   1.1 Document $w \in Doc$
   1.2 Classes $Y = \{y_1, y_2, y_3 \ldots, y_n\}$
   1.3 Training set $T = \{(w_1, y_1), (w_2, y_2), (w_3, y_3) \ldots, (w_n, y_n)\}$
2. Output
   2.1 $classify : Doc \rightarrow Y$
3. What's the downside?

Part II: Representation

Ever seen a word cloud?

What’s a word cloud actually tell you?

Ever seen a word cloud?

What's a word cloud actually tell you?

1. Technical name: Bag of Words

Ever seen a word cloud?

What's a word cloud actually tell you?

1. Technical name: Bag of Words
2. FYI: 'Bag of Words' is also a good insult to call someone

Ever seen a word cloud?

If we created a Bag of Words for the descriptions of the various CMSC courses what might we see?

Term frequency 1

Term frequency 1

1. tf_{ij}: The frequency of word j in document i

Term frequency 1

1. tf_{ij}: The frequency of word j in document i
2. More general than bag of words.

Term frequency 1

1. tf_{ij}: The frequency of word j in document i
2. More general than bag of words.
3. Some adjustments:

Term frequency 1

1. tf_{ij}: The frequency of word j in document i
2. More general than bag of words.
3. Some adjustments:
   3.1 $\log(1 + tf_{ij})$: Reduce impact of outliers

Term frequency 1

1. tf_{ij}: The frequency of word j in document i
2. More general than bag of words.
3. Some adjustments:
   3.1 $\log(1 + tf_{ij})$: Reduce impact of outliers
   3.2 $\frac{tf_{ij}}{max_j tf_{ij}}$: Normalize by most common word

Term frequency 2

Term frequency 2

1. You can use term frequency to train classifiers!

Term frequency 2

1. You can use term frequency to train classifiers!
2. Linear classifiers: "How Dickensian is this novel?"

Term frequency 2

1. You can use term frequency to train classifiers!
2. Linear classifiers: “How Dickensian is this novel?”
3. Think of some examples.

Term frequency 3

Term frequency 3

1. Inverse term frequency is a thing, too!

Term frequency 3

1. Inverse term frequency is a thing, too!
2. What do you think that means?

Term frequency 4

Inverse Term Frequency

$$idf_j = \log(\frac{\#Doc}{\#Doc \ni j})$$

Part III: Advice

NLTK in Python

NLTK in Python

1. I strongly recommend that you do the assigned reading on NLTK

NLTK in Python

1. I strongly recommend that you do the assigned reading on NLTK
2. You don't have to worry about implementing these things!

Thanks for your time!

:)