Text Data

ENSF 444

Discuss how to apply Machine Learning techniques to text data

Chapter 7 in textbook

The types of datasets

Numerical: Contains measurable quantities and can be analyzed mathematically - Temp ,Humidity and test scores

Categorical: Comprises a set of categories or groups - Colors, product categories, and yes/no responses

Time Series: Captures data points at successive time intervals - useful for analyzing trends over time

Image: Consists of image files, often used in computer vision tasks to identify patterns or objects

Text: Collection of words, sentences or documents

Text Data

1. Understanding Text data - Collection of words, sentences or documents , varies in length and complexity
2. Text vs Categorical - text data is more fluid often forming more meaning phrases where categorical does not and only has pre defined features
3. Analyzing Text: Corpus and Documents - using examining a large body of text known as corpus, and within corpus, each individual text entry is a document.

What is a bag of words?

A bag of words is a technique in natural language processing where we ignore the structure of the input text and focus solely on word occurrence's.

Basically holding a bag of words and then counting how many times a word appears

How do we represent this BOW??

1. Tokenization - This process is splitting each document(text) into individual words or tokens -done by parsing whitespaces, punctuation, or other delimiters.
2. Vocabulary Building - Create a **vocabulary** containing all unique words/tokens that appear in any documents - each word is assigned a unique index (usually alphabetical)
3. Encoding - for each document we count how often each word from the vocabulary appears in that document - the resulting vector represents the word frequencies for that document

Heres how we would implement it using scikit

- In the scikit-learn library, the CountVectorizer is utilized to transform text data into a bag-of-words representation.

- The CountVectorizer method standardizes all text data to lowercase, ensuring that words with identical spellings are recognized as the same token.

Enhancing BOW: Stopward Removal

In BOW, some words a common, and they dont have any useful info for the actual content of the document. These are known as stop words, they can be removed to improve the analysis.

2 ways to remove

1. Utilize predefined list of stopwords specific to aa language
2. Excluding words that appear too frequently across the documents

TfidfVectorizer

![[Screenshot 2024-04-15 at 10.30.02 PM.png]]

When implemented, it looks like this

TF-IDF Matrix (1/)

UNIVERSITY OF CALGARY

- Calgary ~~is~~ known ~~for its~~ annual Stampede.
- ~~The~~ Calgary Tower ~~offers~~ stunning views ~~of the~~ city.
- Calgary~~'s~~ weather ~~can be~~ unpredictable.
- Calgary~~,~~ Calgary, ~~a~~ city ~~so~~ vibrant, ~~so~~ vibrant.
- ~~The~~ Rockies~~, the~~ Rockies~~, so~~ majestic~~, so~~ majestic.

- **Doc 1:** ['calgary', 'known', 'annual', 'stampede']
- **Doc 2:** ['calgary', 'tower', 'offers', 'stunning', 'views', 'city']
- **Doc 3:** ['calgary', 'weather', 'unpredictable']
- **Doc 4:** ['calgary', 'calgary', 'city', 'vibrant', 'vibrant']
- **Doc 5:** ['rockies', 'rockies', 'majestic', 'majestic']

Number of Docs = 5
Number of Terms = 14

n-Grams

Expanding Bag-of-Words with n-Grams

UNIVERSITY OF CALGARY

To capture more context, we can extend the BoW model to consider sequences of words:

- **Bigrams**: Pairs of consecutive words.
- **Trigrams**: Triplets of consecutive words.
- **n-Grams**: Sequences of 'n' consecutive words.

```
docs = ["Calgary is known for its annual Stampede.",
        "The Calgary Tower offers stunning views of the city.",
        "Calgary's weather can be unpredictable."
        ]
vect = TfidfVectorizer(ngram_range=(1,2), stop_words = "english")
vect.fit(docs)
tfidf_words = vect.transform(docs)
df = pd.DataFrame(tfidf_words.toarray(), columns=vect.get_feature_names_out())
```

what **ngram_range=(1,2)** means?

- **1**: The lower bound of the range indicates that single words (1-grams or unigrams) will be included.
- **2:** The upper bound of the range indicates that all pairs of consecutive words (2-grams or bigrams) will also be included.

So, with **ngram_range=(1,2)**, the vectorizer will consider both individual words and pairs of consecutive words when creating the vocabulary. This can capture more context compared to just using unigrams.