Data Analysis and Visualization

CentraleDigitalLab@Nice

Deborah Dore - PhD - ddore@i3s.unice.fr
MARIANNE, Université Côte d'Azur, CNRS, INRIA, I3S

Text Data Encoding

Overview

Computers only understand sequences of numbers, but what happens if we want to process a text? **We need to encode it** so that the computer is able to understand it.

Text encoding is defined as the process of converting text into meaningful numeric/vector representations → we still need to preserve context and dependencies between words.

Text Data Encoding

Overview

Text encoding is defined as the process of converting text into meaningful numeric/vector representations → we still need to preserve context and dependencies between words.

Challenges:

1. Ambiguity in meaning
2. High dimensionality
3. Sparsity of data
4. Context and semantics

Text Data Encoding

Overview

Text encoding is defined as the process of converting text into meaningful numeric/vector representations → we still need to preserve context and dependencies between words.

There are various ways to encode text:

1. One Hot Encoding
2. Index Based Encoding
3. Bag of Words (BOW)
4. Term Frequency - Inverse Document Frequency (TF-IDF)
5. Word Embeddings

Text Data Encoding

One Hot Encoding

One Hot Encoding is a method for converting categorical variables into a binary format.

It creates new binary columns (0s and 1s) for each category in the original variable.

Each category in the original column is represented as a separate column, where a value of 1 indicates the presence of that category, and 0 indicates its absence

Text Data Encoding

One Hot Encoding

Example:

Fruit	Price
Apple	10
Mango	20
Mango	17

Text Data Encoding

One Hot Encoding

Example:

Fruit_Mango	Fruit_Apple	Price
0	1	10
1	0	20
1	0	17

Text Data Encoding

One Hot Encoding

Example:

Fruit_Mango	Fruit_Apple	Price
0	1	10
1	0	20
1	0	17

Text Data Encoding

Index Based Encoding

Index Based Encoding is a method for converting categorical variables into a binary format.

The basic idea is to map each word to the index of that word in a dictionary. Each index maps only one word.

Text Data Encoding

Index Based Encoding

Index Based Encoding is a method for converting categorical variables into a binary format.

The basic idea is to map each word to the index of that word in a dictionary. Each index maps only one word.

Example:

“The students of computer science were studying for the exam of computer science”

Text Data Encoding

Index Based Encoding

Example: “The students of computer science were studying for the exam of computer science”

Vocabulary: {The, students, of, computer, science, were, studying, for, exam}

Embedding: [1, 2, 3, 4, 5, 6, 7, 8, 1, 9, 3, 4, 5]

Text Data Encoding

Bag Of Words (BOW)

A **bag-of-words** is a representation of text that describes the occurrence of words within a document. It involves two things:

1. A vocabulary of known words.

2. A measure of the presence of known words.

It is called a “bag” of words, because any information about the order or structure of words in the document is discarded.

The model is only concerned with whether known words occur in the document, not where in the document.

Text Data Encoding

Bag Of Words (BOW)

Example:

> *“It was the best of times,*
> *it was the worst of times,*
> *it was the age of wisdom,*
> *it was the age of foolishness”*

Step 1. Create a vocabulary:

Vocabulary: {it, was, the, best, of, times, worst, age, wisdom, foolishness}

Text Data Encoding

Bag Of Words (BOW)

Vocabulary: {it, was, the, best, of, times, worst, age, wisdom, foolishness}

Step 2. Create document vectors:

The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model. The simplest method is to use one hot encoding.

“It was the best of times” = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

"*it was the worst of times*" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]

"*it was the age of wisdom*" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]

"*it was the age of foolishness*" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

Text Data Encoding

Term Frequency - Inverse Document Frequency

A key issue with bag-of-words is that simply tracking the frequency of words can lead to meaningless words gaining too much influence over a model.

TF-IDF is a method that solves this problem. It consists of two components:

1. **Term Frequency:** notes the frequency of a word in a document
2. **Inverse Document Frequency:** notes the rareness of words across all documents and downplays words that appear across all documents

TF-IDF is more likely to reduce the importance of a word the more times it appears across all documents

Text Data Encoding

Term Frequency - Inverse Document Frequency

TF-IDF is a method that solves this problem. It consists of two components:

1. **Term Frequency:**

$$TF(t, d) = \frac{\text{num of times the term } t \text{ appears in document } d}{\text{total number of terms in document } d}$$

2. **Inverse Document Frequency:**

$$IDF(t, D) = \log \frac{\text{Total number of documents in corpus } D}{\text{Number of documents containing term } t}$$

Text Data Encoding

Term Frequency - Inverse Document Frequency

Example:

- Document 1: *“The cat sat on the mat.”*
- Document 2: *“The dog played in the park.”*
- Document 3: *“Cats and dogs are great pets.”*

Term Frequency for the word **cat**:

- TF(cat, d1) = $\frac{1}{6}$
- TF(cat, d2) = 0
- TF(cat, d3) = $\frac{1}{6}$

Text Data Encoding

Term Frequency - Inverse Document Frequency

Example:

- Document 1: *“The cat sat on the mat.”*
- Document 2: *“The dog played in the park.”*
- Document 3: *“Cats and dogs are great pets.”*

Inverse Document Frequency for the word **cat**:

- IDF(cat, D) = $\log \frac{3}{2} = 0.176$

Text Data Encoding

Term Frequency - Inverse Document Frequency

Example:

- Document 1: *“The cat sat on the mat.”*
- Document 2: *“The dog played in the park.”*
- Document 3: *“Cats and dogs are great pets.”*

TF-IDF (A higher TF-IDF score means the term is more important in that specific document):

- For Document 1 = $TF(\text{cat}, \text{d1}) * IDF(\text{cat}, D) = \frac{1}{6} * 0.176 = 0.029$
- For Document 2 = $TF(\text{cat}, \text{d2}) * IDF(\text{cat}, D) = 0$
- For Document 3 = $TF(\text{cat}, \text{d3}) * IDF(\text{cat}, D) = \frac{1}{6} * 0.176 = 0.029$

Text Data Encoding

Word Embeddings

Word Embeddings are numerical vector representations of words that capture their meanings, relationships, and contexts. Unlike traditional one-hot encoding, which treats words as independent, embeddings map similar words closer together in a high-dimensional space

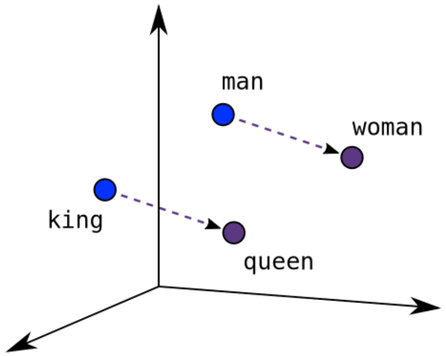
Why do we need word embeddings?

- Words have contextual meanings that one-hot encoding cannot capture
- They help in understanding word relationships and semantic similarities
- Essential for NLP tasks like sentiment analysis, machine translation, and chatbots

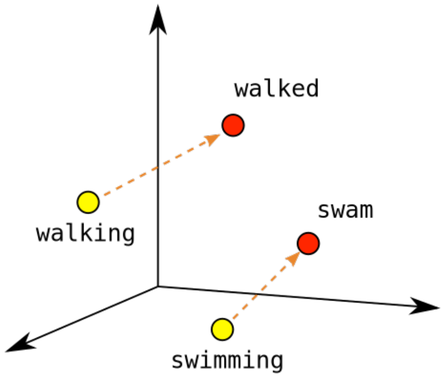
Text Data Encoding

Word Embeddings

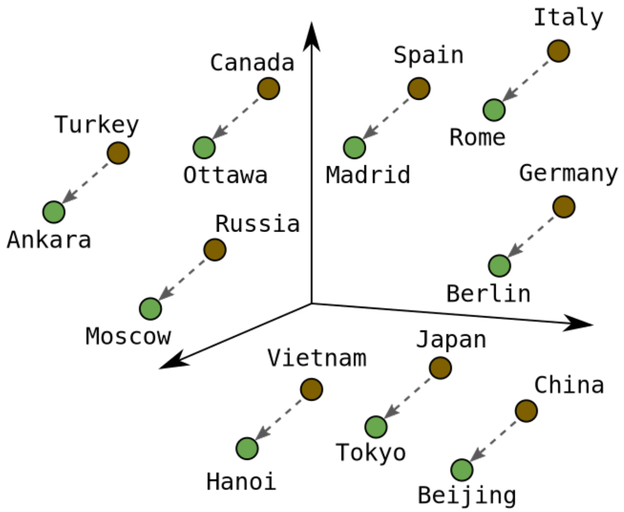
Word Embeddings are numerical vector representations of words that capture their meanings, relationships, and contexts. Unlike traditional one-hot encoding, which treats words as independent, embeddings map similar words closer together in a high-dimensional space



Male-Female



Verb Tense



Country-Capital

Text Data Encoding

Word Embeddings

Word Embeddings:

- **Word2Vec**:
  - **CBOW (Continuous Bag of Words):** Predicts a target word from surrounding context
  - **Skip-gram:** Predicts surrounding words given a target word
- **GloVE**:
  - Captures co-occurrence statistics from a corpus.
  - More robust for general language understanding.

Text Data Encoding

Applications in NLP tasks

Word embeddings enhance various NLP tasks:

- **Sentiment Analysis:** Understanding words like “good” and “great” as similar
- **Machine Translation:** Capturing cross-linguistic relationships
- **Question Answering:** Improving context-based responses
- **Chatbots & AI Assistants**: Understanding user intent

Example:
Google’s **BERT (Bidirectional Encoder Representations from Transformers)** improves contextual word understanding by using embeddings trained on massive datasets

Useful links

- https://machinelearningmastery.com/gentle-introduction-bag-words-model/
- https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

Demo with Notebook_Text_Data_Encoding.ipynb