GenAI: Best Practices

Release 1.0

Wenqiang Feng and Di Zhen

December 07, 2024

CONTENTS

Welcome to our **GenAI: Best Practices**!!! The PDF version can be downloaded from HERE.

CHAPTER

ONE

PREFACE

Chinese proverb

Good tools are prerequisite to the successful execution of a job. – old Chinese proverb

1.1 About

1.1.1 About this book

This is the book for our Generative AI: Best practics [AutoFeatures] API. The PDF version can be downloaded from HERE. **You may download and distribute it. Please beaware, however, that the note contains typos as well as inaccurate or incorrect description.**

The `API` assumes that the reader has a preliminary knowledge of `python` programing and `Linux`. And this document is generated automatically by using sphinx.

1.1.2 About the author

- **Wenqiang Feng**
  - Sr. Data Scientist and PhD in Mathematics
  - University of Tennessee at Knoxville
  - Webpage: http://web.utk.edu/~wfeng1/
  - Email: von198@gmail.com
- **Biography**

  Wenqiang Feng is Data Scientist within DST's Applied Analytics Group. Dr. Feng's responsibilities include providing DST clients with access to cutting-edge skills and technologies, including Big Data analytic solutions, advanced analytic and data enhancement techniques and modeling.

  Dr. Feng has deep analytic expertise in data mining, analytic systems, machine learning algorithms, business intelligence, and applying Big Data tools to strategically solve industry problems in a cross-functional business. Before joining DST, Dr. Feng was an IMA Data Science Fellow at The Institute for Mathematics and its Applications (IMA) at the University of Minnesota. While there, he helped startup companies make marketing decisions based on deep predictive analytics.

Dr. Feng graduated from University of Tennessee, Knoxville, with Ph.D. in Computational Mathematics and Master's degree in Statistics. He also holds Master's degree in Computational Mathematics from Missouri University of Science and Technology (MST) and Master's degree in Applied Mathematics from the University of Science and Technology of China (USTC).

- **Declaration**

  The work of Wenqiang Feng was supported by the IMA, while working at IMA. However, any opinion, finding, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the IMA, UTK and DST.

1.2 Feedback and suggestions

Your comments and suggestions are highly appreciated. I am more than happy to receive corrections, suggestions or feedback through email (Wenqiang Feng: von198@gmail.com) for improvements.

CHAPTER

TWO

PRELIMINARY

In this chapter, we will introduce some math preliminary which is highly used in Generative AI.

2.1 Math Preliminary

2.2 NLP Preliminary

CHAPTER

THREE

WORD AND SENTENCE EMBEDDING

3.1 Bag-of-Word

Bag of Words (BoW) is a simple and widely used text representation technique in natural language processing (NLP). It represents a text (e.g., a document or a sentence) as a collection of words, ignoring grammar, order, and context but keeping their frequency.

Key Features of Bag of Words:

1. **Vocabulary Creation**: - A list of all unique words in the dataset (the "vocabulary") is created. - Each word becomes a feature.
2. **Representation**: - Each document is represented as a vector or a frequency count of words from the vocabulary. - If a word from the vocabulary is present in the document, its count is included in the vector. - Words not present in the document are assigned a count of zero.
3. **Simplicity**: - The method is computationally efficient and straightforward. - However, it ignores the sequence and semantic meaning of the words.

Applications:

- Text Classification
- Sentiment Analysis
- Document Similarity

Limitations:

1. **Context Ignorance**: - BoW does not capture word order or semantics. - For example, "not good" and "good" might appear similar in BoW.
2. **Dimensionality**: - As the vocabulary size increases, the vector representation grows, leading to high-dimensional data.
3. **Sparse Representations**: - Many entries in the vectors might be zeros, leading to sparsity.

3.1.1 One Hot Encoder

3.1.2 CountVectorizer

To overcome these limitations, advanced techniques like **TF-IDF**, **word embeddings** (e.g., Word2Vec, GloVe), and contextual embeddings (e.g., BERT) are often used.

3.2 TF-IDF

3.3 Word2Vec

3.4 GloVE

3.5 Fast Text

3.6 BERT

CHAPTER

FOUR

PROMPT ENGINEERING

CHAPTER

FIVE

RETRIEVAL-AUGMENTED GENERATION

CHAPTER

SIX

MAIN REFERENCE

BIBLIOGRAPHY

[AutoFeatures] Wenqiang Feng and Ming Chen. Python Data Audit Library API, 2019.