Text Dataset Comparison: BoW, Similarity, and Library Performance

Victor Hugo Figueroa
Kristiania University
vifi001@student.kristiania.no

Alin-Eugen Gusanu
Kristiania University
algu008@student.kristiania.no

Hashem Sheikh
Kristiania University
hash011@student.kristiania.no

Jesse Lang
Kristiania University
jela019@student.kristiania.no

1 Introduction

xxx

2 Literature

xxx

3 Methodology

xxx

3.1 Data Pre-processing

3.1.1 Handling missing values

In order to achieve the highest possible accuracy, it is important to analyse the data and pre-process it beforehand. Null values/nan/missing values can be dropped from the dataset or filled by the most probable attribute value. Machine learning algorithms can be applied to predict the best fitting value based on the other features (Kononenko, 2007). This pipeline used the implementation of the KNeighborsRegressor (scikit-learn.org, 2023) algorithm to predict and fill in the missing numerical values in the classification and clustering problem. For the deep learning model, it was sufficient to drop the values since it was only 0.013% of the entire dataset. In the NLP task there were null values replaced by empty strings to work.

3.1.2 Normalization

Libraries like sklearn provide functions to normalise the dataset for a more equal distribution. Normalization is the process of applying a casting of a specific range (e.g., 0 and 1, or between -1 and +1) to the dataset. This method is essential when there are big differences in the arrays of different features. Furthermore, this way of scaling is convenient when the data set does not contain any outliers (Ali et al., 2014).

Figure X visualizes the normalization process and the necessary steps to cast it to the range 0, 1. To get the normalized value, the equation $x' = \frac{x-\min}{\max-\min}$ can be used, where x represents the value before and x' after normalization. The other two values are the minimum and maximum values of the given feature array.

In order to normalize them to the range -1, +1 (see Figure X), the formula has to be adjusted to $x' = 2\left(\frac{x-\min}{\max-\min}\right) - 1$ (Ali et al., 2014).

3.2 Machine Learning

Machine Learning is split into three main categories, which are supervised, unsupervised and reinforcement learning. This report required the first two classes, where supervised learning is about the predicting of values with regression models, as well as classifying data with predefined labels. On the other hand, there is unsupervised learning which contains the analysis of patterns and can form clusters out of unlabelled data.

3.2.1 Classification

There are several different classification models and each of them fits a specific use case best. The models need to be evaluated and compared to one and another, to find the optimal algorithm. This report analyses seven different models from sklearn (Logistic Regression, Decision Tree, Naive Bayes, Support Vector Machine, Random Forest, XGBoost, KNeighborsClassifier) and evaluates them based on the run time and accuracy. Two functions, one to find the best random state on the train-test-split data, and the other to get the ideal hyperparameters based

on a grid search, loop through the previous defined models and automate the process, when finalised it returns the best score and run time duration.

K-nearest neighbour - KNN

The K-nearest neighbour algorithm is one of the finest examples of instance-based learning. Additionally, it is easy to understand and a simple method for classification problems. Despite its simplicity, it has the capability to yield results that are highly competitive. Not only is it well suitable for classifications but it also fits the requirements for regression predictions (Sen et al., 2020).

The algorithm stores all the given data points and predicts the target based on giving attention to the similarity measurements of the surrounding neighbours in likelihood. The number of neighbours that will be taken in consideration is defined by the "k" variable. Assuming k equals 3, a circular region with the new data point at its centroid is created to encompass only the three closest neighbouring data points on the plane. The determination of the label for the new data point is then based on the distances between the data point and each of its neighbours (Sen et al., 2020).

Some of the advantages are that it handles noisy and large training data well, besides the simplicity of the implementation. A significant limitation of this algorithm arises from the necessity to recalculate the distances from K neighbours for every new instance, resulting in substantial computational time consumption. Additionally, accurately determining the value of K is crucial to achieve a lower error rate (Sen et al., 2020).

Support vector machine - SVM

Another supervised algorithm is the support vector machine. It can handle both, classification, and regression problems, though is it more seen for classification. Furthermore, it can manage numerous instances that involve both continuous and categorical data (Sen et al., 2020).

The algorithm can be defined like following. Items of the dataset with "n" features will be characterised and plotted as points in an n-dimensional space split into classes by a hyperplane with the widest possible margin. The data points are then mapped into the previous defined space to predict their label based on their position relative to the hyperplane (Sen et al., 2020).

A significant performance boost can be seen, when the variable "n" exceeds the total size of sample set. Therefore, is this algorithm mostly taken under consideration for high-dimensional data. Further improvements in performance can be achieved by having a well-constructed hyperplane. Despite its advantages, is a relatively high training time one of its drawbacks. Which leads to slower predictions, especially with large datasets (Sen et al., 2020).

3.2.2 Clustering

xxx

3.2.3 Deep Learning

xxx

3.3 Natural Language Processing

xxx

3.3.1 Text Processing and Feature Extraction

xxx

3.3.2 Topic Modelling

xxx

3.3.3 Searching for Similar Movies

xxx

4 Analysis

xxx

5 Conclusion

xxx

References

Bird, S., Klein, E. & Loper, E. (2009), *Natural Language Processing with Python*, O'Reilly Media, Inc.

Harris, C., Millman, K. & van der Walt, S. e. a. (2020), 'Array programming with numpy'.

Lane, H. (2019), *Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python*, 1st edn, Manning.

Matplotlib—Visualization with Python (n.d.), `https://matplotlib.org/`. Retrieved 28 October 2023.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. & Duchesnay, E. (2011), 'Scikit-learn: Machine learning in python', *Journal of Machine Learning Research* .

Python Data Analysis Library (n.d.), `https://pandas.pydata.org`. Retrieved 28 October 2023.