HERD Software User Manual

Introduction

This manual is a guide to the usage of our HERD software. The purpose of the HERD software is to build predictive models from which to make predictions on the HERD Purpose and HERD Field categories. We provide a fully trained model which can be used to make predictions. In addition, we supply a software mechanism for training new models on our own data.

For those with some familiarity with python and machine learning, we provide our complete source code and documentation. We believe that this will allow easy modification of the code for alternative uses or applications.

Software Installation

Linux

Use of the code in Linux operating systems varies from distribution to distribution. Python 3.4 (or greater) is required to run the code. This may be installed via your operating systems package manager (e.g. Fedora: yum/dnf, Ubuntu: apt-get install, etc).

Windows

We recommend installing the Windows scientific python package WinPython to provide a working Python 3.4 (or greater) installation.

TODO: Add more details on how to run on Windows.

Software Description

General Overview

The software package provides two utilities or tools which provides the ability to train a model on data, or to use a model to make predictions on new data. The programs accept files with the ".xlsx" extension, which may be generated from either Microsoft Office, Open Office, or LibreOffice.

Data is goes through two stages of processing, a vectorization stage followed by a model stage. In the vectorization stage, the input data is read in as plain text and converted to a vector (numerical) representation. This is a necessary step, as the model is unable to read in and understand text directly.

In the model stage, the model uses the data for training (learning) or to generate predictions. The function (training or prediction) of the model is dependent on which software utility/tool is called. Examples of these functions are detailed later in the document.

Model Configuration

The software allows independent configuration of the vectorization and model stages. The configuration file contains detailed instructions on how the the vectorization and modeling stages are built. The configuration file for a model and the model itself is stored in separate files.

This configuration file is expected to be a human-readable YAML formatted file. For information on the YAML format please see http://yaml.org/. Please note that the spacing in the configuration file is extremely important, and that replacing spaces with <tab> will render the configuration file unreadable.

The configuration file is divided into four sections:

- purpose_vectorizer
- field_vectorizer
- purpose_model
- field_model

As can be seen, separate configurations are used between the purpose and field categories in both the vectorization and modeling stages.

Provided Model & Configuration

Included with the software, we provide a configuration file, and a model file. These files are a pre-trained model which was trained on our data at KU. This allows a user to use our model to generate predictions for their data. Alternatively, one could use the provided configuration file to train a model with the same parameters, but specifically tailored for their data.

Please note that the performance of the model is highly dependent on the amount of data which it is trained on. Training a model on a small number of samples (less than several hundred) will likely cause the model to "memorize" the data rather than learn anything useful from it. Ideally, the number of training examples should be in number of several thousand. Though in our case we have found the models useful with as few as around 1500 examples.

Provided Software Utilities

We provide two high-level command-line utilities. These require a working Python3 installation, and the installation of any dependent software libraries. The utilities are designed to be simple to use, one utility is concerned with model training, and the other with generating predictions.

Model Training

Models are trained using the "trainModel" utility in the "Tools" package. The trainModel utility is called in the following manner:

```
trainModel --data <data_path> --config <config_path> --model <model_path>
```

The description of the options are as follows:

- INPUT: <data_path>, is the path to the input data file for training a model. The input is to be an excel spreadsheet. It is expected to contain the following columns:
    - "sow" – The statement of work for the funding source, this is the abstract of the proposal.
    - "purpose" – The assigned purpose category.
    - "field" – The assigned field category.
- INPUT: <config_path>, is the path to the configuration file for training the model. This file contains details on how to vectorize the data, the type of model to train, and specific tunable parameters controlling model behavior.

- OUTPUT: <model_path>, this is the path determining where to store the trained model.

Example of running the program:

python3 Herd_v2.Tools.trainModel --data trainData.xlsx --config model.config --model myModel.model

Naturally, you must have python3 and the necessary python libraries installed, as well as having the HERD package added to your python path.

Generating Predictions

A second utility we provide is one to make predictions on new data using an already trained and saved model. This utility is called in the following manner:

makePredictions --model <model_path> --data <data_path> --output <output_path>

The descriptions of the required options are as follows:

- INPUT: <model_path>, is the path to the model file used to make predictions on the data.
- INPUT: <data_path>, is the path to the Excel data file containing the data with which to make predictions on. It is expected to contain the following columns:
    - "sow" – The statement of work for the funding source, this is the abstract of the proposal.
- OUTPUT: <output_path>, is the output file to be created with which to store predictions in. The first spreadsheet page is named "Input" and will contain the same data as the first page of the input data spreadsheet. A second spreadsheet page is created called "Predictions", this page will contain three columns; "sow, "purpose", and "field". These columns contain the statement of work, and the predictions labels for the purpose and field categories.

Example of running the program:

python3 Herd_v2.Tools.makePredictions --model myModel.model --data predictData.xlsx --output predictions.xlsx

Additional Provided Utilities and Software Libraries

For those familiar with Python programming and machine learning, we provide our complete code base. This includes utilities and libraries for conducting a grid search over different parameters, estimating classifier generalization, and evaluating different models. The code is well commented and contains docstrings which are readable by the python documentation generation utility called Sphinx. Generated Sphinx documentation is provided in both HTML and PDF formats.