BRIEFING NOTE: MACHINE LEARNING PILOT
INFERRING HXL HASHTAGS FOR TABULAR DATA

BY MICHAEL RANS
Data Scientist
Centre for Humanitarian Data

Summary

A key challenge for the humanitarian community is creating data that is interoperable among dozens of organisations and across different crises. The Humanitarian Exchange Language (HXL) is a simple data standard for spreadsheets, web APIs, and databases that is designed to solve this problem. While the process of adding HXL hashtags to tabular data is straightforward, there is some effort involved in determining which ones to use at the top of columns.

The Humanitarian Data Exchange (HDX) now contains a substantial number of 'HXLated' datasets (over 3,000 as of October 2018). This pilot considered whether these datasets could be used to train the HDX system to make accurate hashtag predictions on new datasets that are added to the platform. This would save data contributors from having to add the tags and could result in exponential growth of interoperable data.

The Current Manual Process

In a HXLated file, each column of data has a hashtag indicating the type of data contained in that column such as country codes, population counts or clusters. Additionally, each column may have one or several attributes that give further information about the data. Assigning HXL tags to datasets is a manual task requiring a data manager to label each column with the hashtag he or she finds the most appropriate.

This process is summarised below:

1. Grab a **spreadsheet** of humanitarian data.

2. Insert a **new row** between the column header and the data.

3. Add **HXL hashtags** or get advice on what to use in the **HXL Tag Assist**[1].

Similarly, the creator of an API would have to identify manually what HXL hashtags to include.

Proof of Concept HXL Hashtag Prediction

During the summer of 2018, an HDX volunteer Data Scientist, Henrik Sjökvist, began an investigation on the feasibility of predicting hashtags in new data based on information inferred from previously tagged datasets. He produced a proof of concept consisting of three main components which together form a pipeline from raw data to hashtag predictions. These include:

1. Data cleaning
2. Feature extraction
3. Modelling

The steps outlined above were implemented in the programming language Python. The code is available on GitHub as a series of independent scripts.

1. Data cleaning

The raw data used in this project is an extract of all column headers from all datasets marked as containing HXL hashtags. The column headers are strings containing textual descriptions of the data contained in each column and may be in plain English, e.g. "country name" or some other less understandable form. Hence, we need to do some light data cleaning.

To demonstrate how the preprocess.py script works, let's take the made up header text "ThisIsMy, SUPER, Header" and work with it in the following steps:

1. Stringify: certain datatypes may have been interpreted as non-string datatypes and are converted to strings ("ThisIsMy, SUPER, Header" is already a string).

2. Split on punctuation characters: some words are separated by punctuation characters which are replaced with whitespace ("ThisIsMy SUPER Header").

3. Lowercase those words which are all uppercase: although we want all characters to be in lowercase, converting immediately could destroy useful information. A common format for headers on HDX consists of capitalising the first letter in each word, e.g. "ThisIsASampleSentence". If we lowercase this header into "thisisasamplesentence", this will make it very difficult to separate into individual words. Instead, we want to first split on each new capital letter, but we must be careful because words which are entirely uppercase would be divided into individual characters. Hence, we only lowercase words which are all capitals ("ThisIsMy super Header").

4. Split words on uppercase characters: we add a whitespace before each capital letter ("This Is My super Header").

5. Remove excess whitespace: the original text and the string manipulations performed upon it may lead to excess whitespace which we remove ("This Is My super Header").

6. Lowercase everything: replace capitals with lowercase ("this is my super header").

2. Feature extraction

Now that we have cleaned column headers, we want to extract features from them. Typically, machine learning models take numerical data as input. In order to make use of standard classification algorithms, we must find a way to represent

[1] https://tools.humdata.org/examples/hxl/

the column headers as numbers rather than strings. Of the different methods that were tried, the most promising turned out to be word embeddings in which words are mapped to vectors of numbers.

The major advantages are:

- Word embeddings are typically of lower dimension.
- Their size does not increase with the amount of data.
- Without being programmed to know about such concepts as gender, word embedding models are able to capture deep semantic language patterns as linear vector relations, for example: “king” + “woman” – “man” ≈ “queen”.

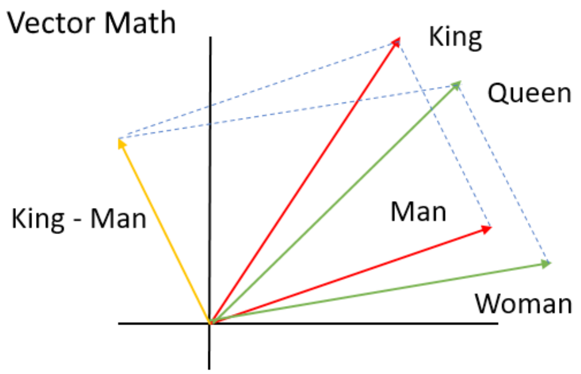


Figure 1: Semantic language patterns captured using word embeddings

The open source model selected for this project, called fastText, was developed by Facebook’s AI research team. It is a neural network model which requires enormous amounts of text data in order to train high quality word embeddings. Fortunately, the developers have published pretrained models for anyone to use.

The script extract_features.py uses fastText to convert each column header into a single vector by first obtaining a word embedding for each word in the header and then computing the vector mean of all the resulting word embeddings. The resulting vector consists of 300 floating point numbers between -1 and 1.

3. Modelling

Since the purpose of this project is to match each column header to one HXL hashtag (like matching an image of a fruit to either an apple, orange or pear), we must use a multiclass (singlelabel) classifier. The best results were consistently obtained using a Multilayer Perceptron (MLP) classifier which implements a supervised learning algorithm that learns a non-linear function by training on a dataset.

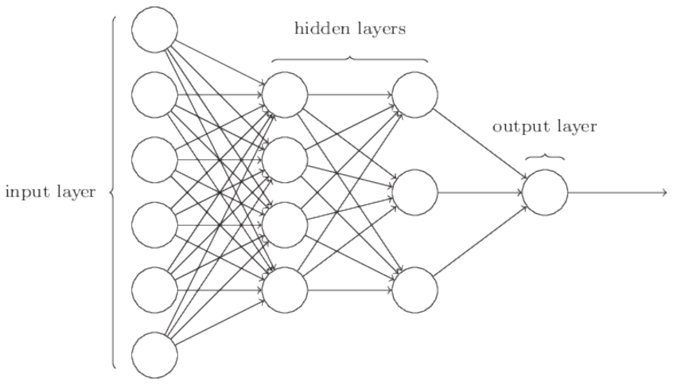


Figure 2: A multilayer perceptron classifier

The classifier has several parameters that can be tuned to improve performance, a process carried out in the program MLP_parameter_gridsearch.py. Given parameters to tune and a range of values to explore for each one, it tests each combination of different parameter values and outputs the set which yield the highest classification accuracy.

When a set of tuned parameters have been found, a final classification model can be trained and evaluated. The script MLP_train_classifier.py trains an MLP classifier and outputs the trained model and a confusion matrix for performance analysis.

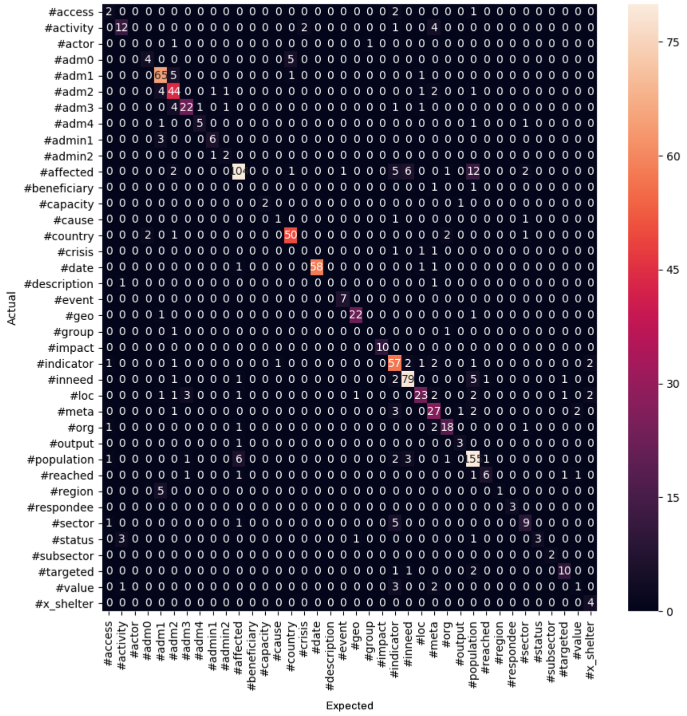
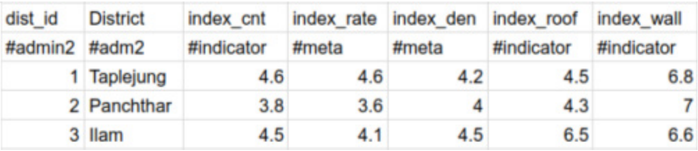


Figure 3: Confusion matrix depicting classification accuracy

We can now take the model and use it to predict HXL hashtags for new untagged datasets. The program tag_new_dataset.py takes the trained model output by the previous script and uses it to infer tags for a new dataset, saving it with an appended row containing the predicted hashtags.

Results

This project has helped to demonstrate the feasibility of inferring HXL hashtags for certain column headers. The best model so far achieves a classification accuracy slightly above 80% on a third of the test data. An example of the model in action is shown below.



dist_id	District	index_cnt	index_rate	index_den	index_roof	index_wall
#admin2	#adm2	#indicator	#meta	#meta	#indicator	#indicator
1	Taplejung	4.6	4.6	4.2	4.5	6.8
2	Panchthar	3.8	3.6	4	4.3	7
3	Ilam	4.5	4.1	4.5	6.5	6.6

Figure 4: Hashtag prediction applied to an untagged dataset

The table shows how this attempt at inferring hashtags has been partially successful. #adm2 and #indicator are correct. #admin2 is almost right -- the training data based on HXLated datasets in HDX contains this misspelled hashtag which should be #adm2. #meta should be #indicator.

Future Directions

More accurate predictions will require utilising other features besides the header text. It is easy to tag a column with the header "Date" with "#date", but in many cases it is not possible to give a confident prediction of any hashtag based on the column header alone.

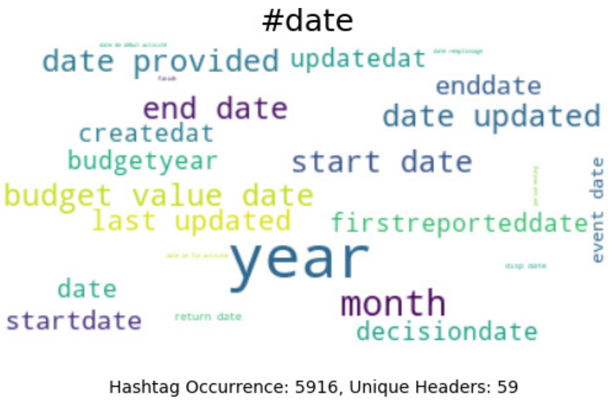


Figure 5: Word cloud of the column headers corresponding to #date

1. More features for the classifier

Fortunately, there are other sources of information in an HDX dataset that could be used to craft more features for the classifier. These include but are not limited to:

- The actual data content of a column
- The position of the column in the dataset
- Adjacent columns
- Organisation supplying the data
- Location of the data

Specifically, the performance of the model would be improved significantly by the inclusion of information about a column's datatype and its adjacent columns.

2. Predicting attributes

Another future path is to attempt to predict HXL attributes. This is a more difficult task as it is a multilabel problem. While there are classifiers which can handle this, a challenge is to distinguish between hashtags and attributes. A column header can have zero or more attributes but must have exactly one hashtag.

One solution is to train separate models for hashtag and attributes, inferring the hashtag and passing it in as an extra feature to the attribute model. At this stage, there are not enough HDX datasets that include hashtags with attributes. Achieving high accuracy in a multilabel problem requires much more training data.

3. Adding a confidence level

Rather than just outputting a hashtag, it would be good to have a confidence level as well. A future user interface could have several hashtag suggestions with the default being the one with the highest confidence level. This would require that the model be adapted to produce more than one result per column header.

Conclusion

We believe adding machine learning to the HXL tagging process can drive adoption of the HXL standard resulting in more interoperable humanitarian data. Rather than aiming for a fully automated system that runs the risk of occasionally choosing the wrong hashtag, we envisage that the ultimate decision will be left in the hands of the domain experts -- the data contributors. We see the technology as an assistant, complementing rather than replacing specialist knowledge. We are excited at the potential of this work and we welcome ideas and suggestions for improvements. Contact us at centrehumdata@un.org.