Machine Learning for Humanitarian Data

Tag Prediction using the HXL Standard

Vinitra Swamy[1], Elisa Chen[2], Anish Vankayalapati[2], Abhay Aggarwal[2], Chloe Liu[2], Vani Mandava[1], Simon Johnson[3]

Microsoft AI & Research[1], UC Berkeley Division of Data Sciences[2], UN Office for the Coordination of Humanitarian Affairs[3]

Motivation

- There is a need for **data interoperability and standardization** for humanitarian data
- However, labor-intensive process of data labeling requires **crisis responders to spend hours wrangling data** instead of assisting with relief efforts

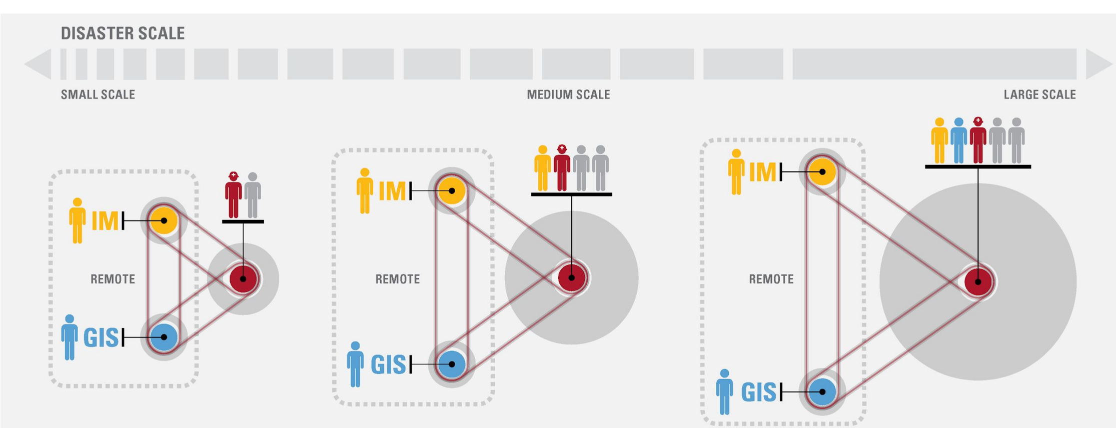

Fig.1 Red Cross disaster response team structure -- 2 remote information specialists coordinate with a team lead and first responders

American Red Cross

- We propose a **deep learning solution** to improve efficiency in crisis response

Project Goals

- The Humanitarian Data Exchange (HDX), an open data platform to store humanitarian datasets, uses the Humanitarian Exchange Language (**HXL**) to tag data columns
- The goal for the collaborative project is to build a machine learning model to add HXL header tags to over **6,000** untagged datasets, as well as create **a tool to predict tags in real-time** for new unseen datasets.

HXL Standard

Fig.2 HXL Postcard providing an overview of the HXL standard by topics

Data Cleaning and Scraping

Data Cleaning, Web Scraping, Preprocessing

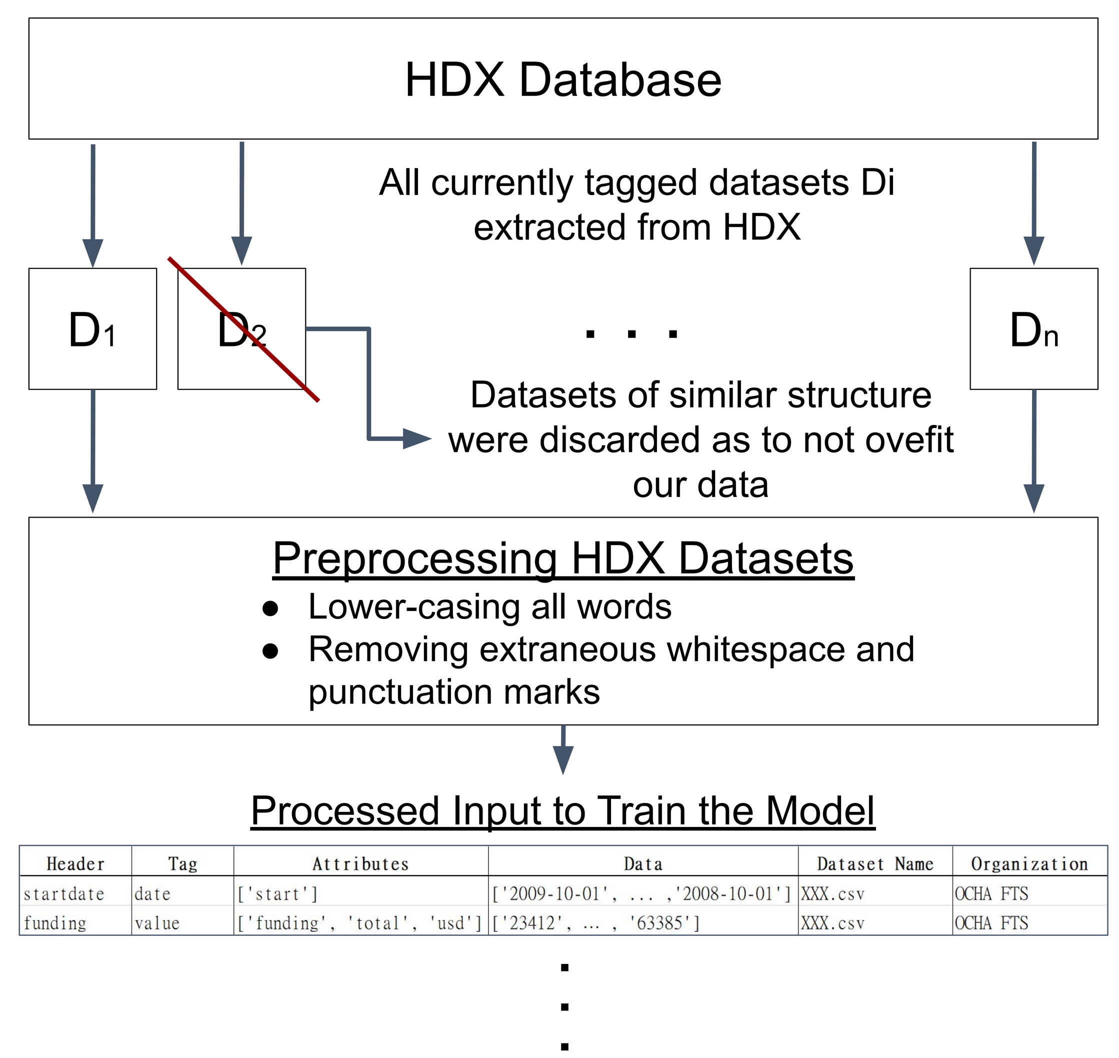

Fig.3 Preprocessing pipeline to transform raw input datasets from HDX into training dataset

Tag Prediction Pipeline

Multilayer Perceptron Classifier (MLP), FastText Embeddings

Model Design

- The model was trained with our processed input using **headers, first seven rows of data and organization name** as features
- We employed MultiLayer Perceptron (MLP) classifier with ReLU activation layers, a learning rate of **0.001**, epsilon of **1e-08** and hidden layers of size **150**

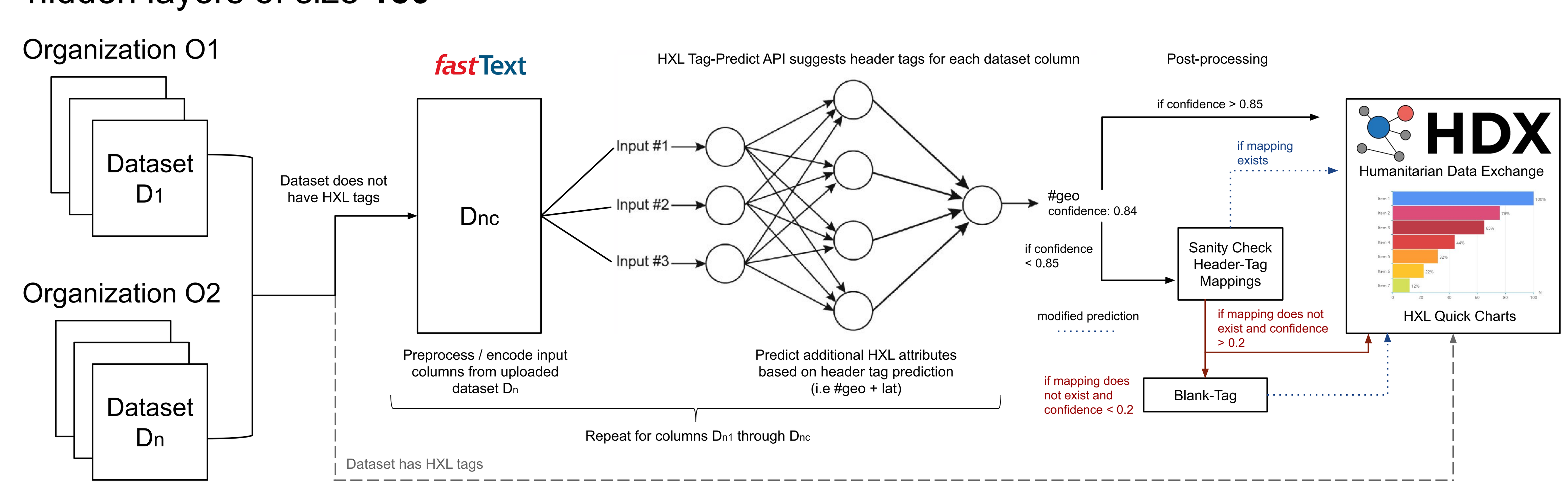

Fig.4 Tag Prediction Pipeline: the classification architecture contains preprocessing, encoding, modeling, and postprocessing sections

- The model was trained and tested against a test size of **33%** of the input sample of **3659** datasets

Word Featurization and Encoding

- **FastText**, a Facebook AI Research (FAIR) library for efficient learning of word representations, was used to **encode the features** and stack input word embeddings

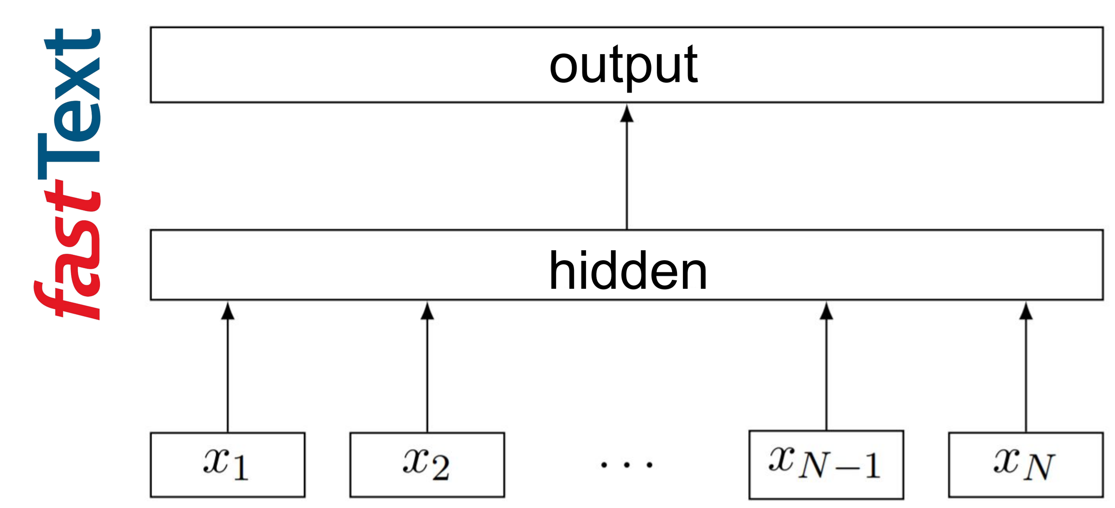

Fig.5 Model Architecture of FastText for a sentence with N n-gram features. The features are embedded and averaged to form the hidden variable.

- FastText uses **a hierarchical softmax** that takes advantage of the unbalanced distribution of the classes to speed up computation
- Similar to Word2Vec, FastText uses Continuous Bag-of-Words and Skip-Gram for word representation learning, but instead of feeding individual words into the neural network, FastText breaks words into several n-grams allowing **rare words to be represented more effectively as well**

Results

- Using the header, organization name and the data content as features in the model, we obtained an **accuracy of 94.3%** using the MLP Classifier on the holdout test set.
- We perceive a roughly **15% improvement** compared to the previous model which had an ~80% accuracy on its test set.
- The accuracy of the model was tested against multiple Sci-Kit Learn classification algorithm implementations and reached the highest accuracy with the MLP Classifier

Classifier	Accuracy	Hyperparameters
K Nearest Neighbors	90.4%	K = 3
MLP Classifier	94.3%	hidden layers = 150
Random Forest Classifier	67.7%	max-depth = 5
Naive-Bayes Classifier	62.5%	

Fig.6 Tag-Prediction Model accuracy by classifier

Continuing and Future Work

- **Data skew**: We expect accuracy to be lower on tags that do not appear in the training set
- **Mitigate false predictions:** Postprocessing in which we compare predictions with weak confidence levels against common header-tag mappings
- **HXL Dashboard Integration:** HXL tags enable the development of an UI for graphical data comparison, quick analysis and interoperability
- **Attribute Prediction Model:** Improvement on a follow-up model to predict the additional HXL attributes for each header
- **ONNX: Open Neural Network eXchange** Exporting and continuing to optimize our model for real time inference performance using the ONNX model format

Acknowledgements