# Capstone Project Roman Levitas

Machine Learning Engineer Nanodegree April 24, 2017

**Text Classification on Stack Exchange Questions**

# Definition

**Overview**

The aim of this investigation will be to build a text classifier, but the twist is that the classifier will be trained on topics independent of the testing set, making this a classic Natural Language Processing (NLP) problem with a transferred learning approach.

The data to be analyzed is a subset of the Stack Exchange Data Dump[[1]](#footnote-1) published on December 15, 2016 focusing on 6 topics in particular: biology, cooking, cryptography, diy, robotics, and travel. The testing set is a 7th topic--physics.

In effect, the classifier to be designed will have knowledge of 6 seemingly independent topics. The idea of there being a common thread or unifying theme if looked at from the perspective of physics among these topics is the goal of this investigation.

The history of Stack Exchange's question-and-answer format dates back to as recent as 2008 when StackOverflow was created, the originating leg of the Stack Exchange network, which allows users to crowdsource knowledge on the topic of computer science/software engineering.

The data dump contains the titles, text, (the input set) and tags (which is what we will be predicting via the classifier) of Stack Exchange questions in the form of a comma separated value (CSV) document.

**Problem Statement**

The problem at hand is defined by the Kaggle team's competition title: 'Transfer Learning on Stack Exchange Tags' [[2]](#footnote-2) which aims to "Predict tags from models trained on unrelated topics". Specifically, predicting tags for physics questions after training the classifier on questions provided in the 6 different fields mentioned above.

Underlying in this approach is the presumption that physics is the unifying concept, so the investigation is itself an exercise into a problem without a definite answer but if there is a correlation that can be found, it can certainly shed light and illuminate the grey area in question: Is Physics at the heart of eveything?

Predictions can be compared against the physics questions' actual tags and thus the model can be ranked on the correctness of its categorization using metrics discussed below.

**Metrics**

The evaluation metric for this competition is Mean F1-Score[[3]](#footnote-3). The F1 score measures accuracy using the statistical notions of precision (p) and recall (r). Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives to all actual positives (tp + fn).

The F1 score is given by:

where:

and

In the multi-class and multi-label case, Mean F1-Score is the weighted average of the F1 score of each class. [[4]](#footnote-4)

To better understand this, an example is given below:

On the left is an array of predicted tags and on the right are the correct, assigned tags.

Prediction => Actual

[quantum-mechanics, electron, current] =>[ electron, ampere, quantum-mechanics, voltage]

Correct predictions are:

quantum-mechanics:

electron:

while the rest are either false negatives or false positives:

current:

ampere:

voltage:

2 , 1, 2

p = 2/3 , r = 1/2

F1= 4/7 = 0.57

Analysis

**Data Exploration**

The input dataset is 6 separate CSV files for biology, cooking, cryptography, diy, robotics, and travel[[5]](#footnote-5). Each row of the data contains the title, text, and associated tags of a question. The export of the data from StackExchange supported html markdown for the text column so there will be a prerequisite data cleaning step to take into account markdown formatting and html tags/elements.

The first five rows of the biology CSV are shown:



The training set contains ~25,000 entries for diy, ~8,600 for biology, ~10,400 for cryptography, ~2,700 for robotics, ~12,000 for travel, ~15,000 for cooking.

The testing data has ~82,000 entries on physics but lacks the tags, which are to be predicted, submitted to kaggle for verification, and scored.

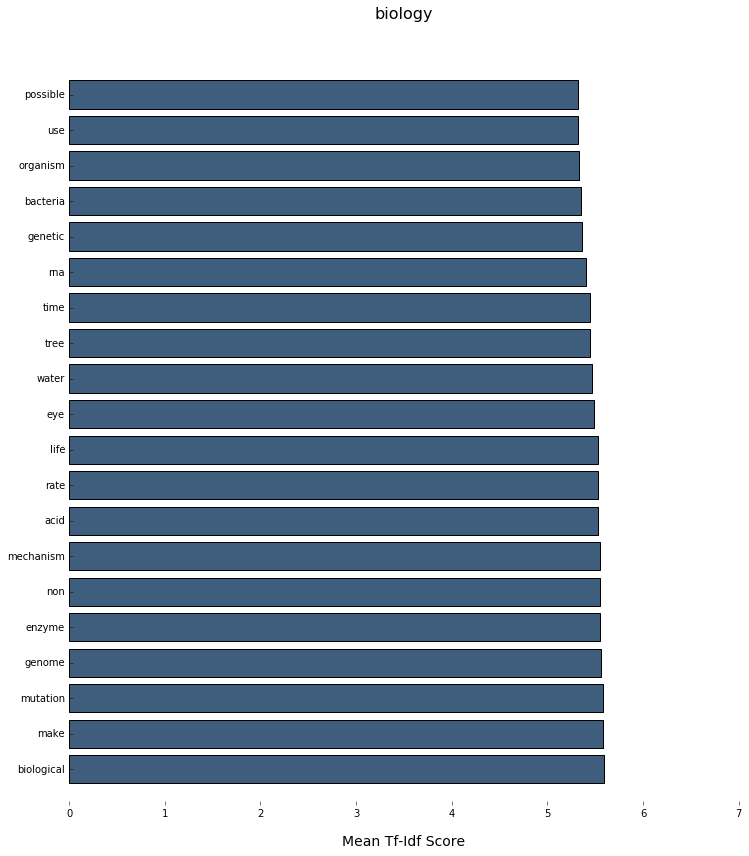
**Exploratory Visualization**

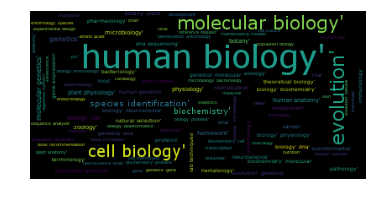
One measure of how important a word in a text collection is its term frequency (tf), how frequently a word occurs in a document. There are, however, words in a document that occur many times but may not be significant; in English, these are words like “the”, “is”, “of”, etc.

Another approach is to look at a term’s inverse document frequency (), which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. This can be combined with term frequency to calculate a term’s -,

which is a way to score the importance of words in a document based on how frequently it appears across multiple documents, or in this case, rows/questions.

To get a sense of the questions, a histogram of the top 20 terms according to their - scoring of the titles from the biology CSV provides some intuition as to what the classifier will be working with as input values, after a data cleaning step.





ASDFA

1. 1. archive.org/details/stackexchange

   [↑](#footnote-ref-1)
2. 1. kaggle.com/c/transfer-learning-on-stack-exchange-tags

   [↑](#footnote-ref-2)
3. 3 kaggle.com/c/transfer-learning-on-stack-exchange-tags#evaluation [↑](#footnote-ref-3)
4. scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\_score.html [↑](#footnote-ref-4)
5. kaggle.com/c/transfer-learning-on-stack-exchange-tags/data [↑](#footnote-ref-5)