Taxonomy Creation

Solution Submission

CT/DT Number:	**CT20192830706**
Contestant Name:	**Tanmay Vijay**
College Name:	**UNIVERSITY COLLEGE OF ENGINEERING, RAJASTHAN TECHNICAL UNIVERSITY, KOTA**

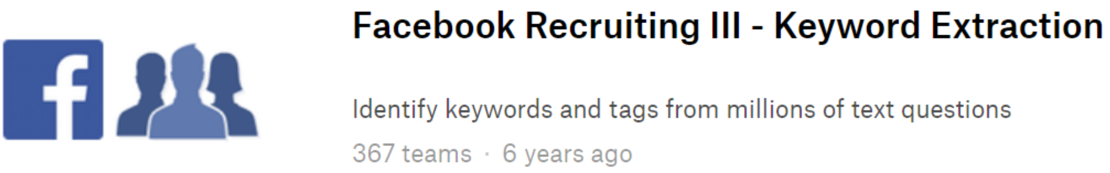
1. Background

For TCS HumAIn 2019, I have chosen the problem of **Taxonomy Creation** from the given problem set.

In Taxonomy Creation problem, we have a dataset collected from some QnA website which contains Title, Body and associated Tags. We have to learn a relationship between Question text ang associated Tags so as to create an automatic tagging system for websites.
In simpler terms, our model has to categorize text into various domains, based on occurrence of some certain words.

Any question can be categorized into multiple domains, that is it can have multiple tags.

Background - This problem and it's dataset is taken from **Facebook Recruiting III - Keyword Extraction** competition held on Kaggle (https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/rules).



Facebook Recruiting III - Keyword Extraction

Identify keywords and tags from millions of text questions

367 teams · 6 years ago

This competition tests the proficiency of a data scientist in regards of his/her analytical skills and creative problem-solving skills.

It examines the data scientist for:

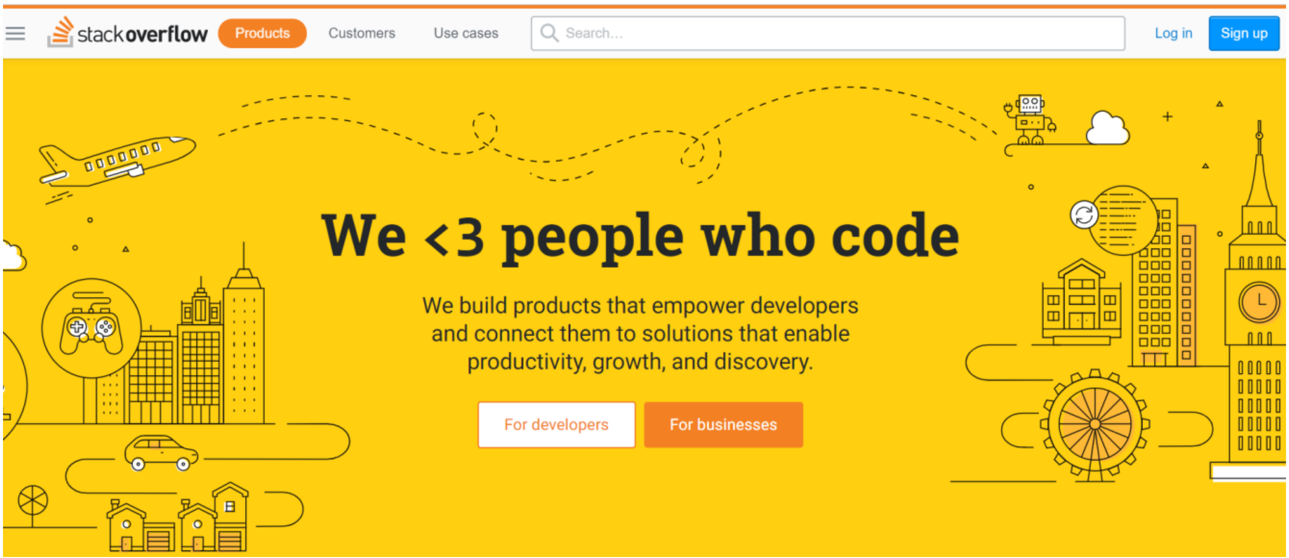
- Skills to tackle large datasets
- Knowledge of Natural Language Processing tasks
- Feature Selection and Feature Engineering skills

Skills required to tackle the problem:

- Machine Learning
- Natural Language Processing
- Data Engineering
- Working with multiple processors

2. My Understanding

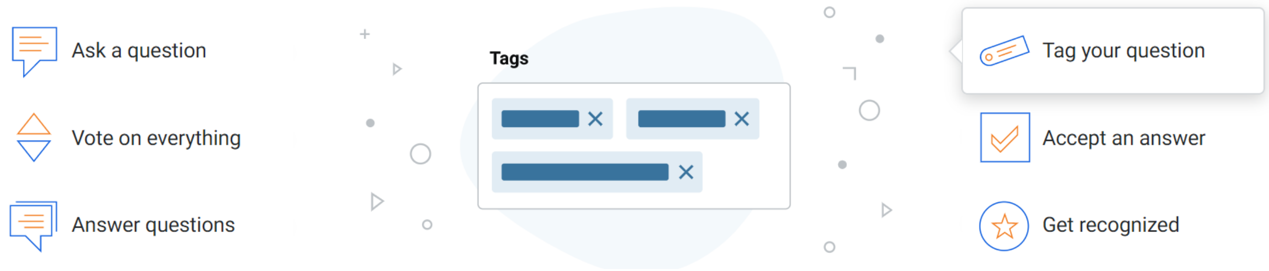
Taxonomy Creation is a very popular Machine Learning problem.


stackoverflow
Products
Customers
Use cases
Search...
Log in
Sign up
We <3 people who code
We build products that empower developers and connect them to solutions that enable productivity, growth, and discovery.
For developers
For businesses

QnA we sites are most useful areas of internet, where we can ask a question and get answers from domain experts.

To aid the users to find related questions and questions they can answer, websites employ a Tagging system, which categorizes questions into their domain and specific experts can now easily find their discipline.

But tagging each question manually is not efficient and waste of crucial human resources.



So, here, we have to create an automatic Tagging System, which can categorize questions into various disciplines based on the text and keywords in that question.

For solving this problem, we're a given a dataset, taken from Facebook Recruiting III competition on Kaggle and created using StackOverflow Questions.

From a Data Scientist's point of view, this is a **Multi-Class Multi-Label** problem, since we have to determine all the classes of tags a question belongs to.
A relationship needs to be established between Question and Tags.

Thus, we need to train model to predict a binary output for each tag for any given question.

Training Dataset contains roughly 60M examples in 4 Columns:

- Id – Unique identifier for each question
- Title – The question's title
- Body – The body of the question
- Tags – The tags associated with the question

Title and *Body* columns contain the text part as strings, *Tags* column contain all the tags concatenated into a string separated by spaces.

3. Scope – Elements being considered in the Solution

From the Dataset, we only use 3 columns:

- Title
- Body
- Tags

Elements used in Code:

- Regular Expression
- NLTK (Natural Language Tool Kit)
- Bag of Word Model
- Counter array
- Data Frames
- Count Vectorizers
- Sparse Matrices

The solution is based on Python Programming Language and extensively use various 3rd party libraries.

Libraries used:

- Numpy – To handle efficient arrays in RAM.
- Pandas – To created manageable Data Frames.
- Scikit-Learn – To apply ML models and algorithms on data.
- Scipy – To store sparse matrices in efficient format.
- Pickle – To save models and vectorizers used on data.

4. Out of Scope – Elements not being considered in the Solution

From the dataset, we ignore the column: **Id**

- Less frequent tags are discarded.
- Examples not containing only top 500 tags are discarded.

Since the dataset size is very large for local machines:

- **We only use the most frequent 500 tags from the dataset.**

We randomly sample examples from dataset such that their associated tags belong to only the top 500 tags.

- **We only use 500,000 examples all which contain tags belonging to the set of top 500 tags.**

This sampling is done to reduce the overhead on the processor and also it can be verified that top 500 tags cover most of the questions and 500,000 quality examples are enough to give highly acceptable output.

5. Assumptions

General Assumptions:

- The application is handled and tested by technical users.
- We only need to train the model and create code to test the model.
- We don’t need to create UI.

Technical Assumptions:

- The CPU and RAM usage are limited to normal household devices.
- Examples in the training which contains no tags are of no significant value and can be ignored.
- Distribution of training data is not biased.
- There is no limitation on CPU and RAM usage during testing.

6. Solution Approach

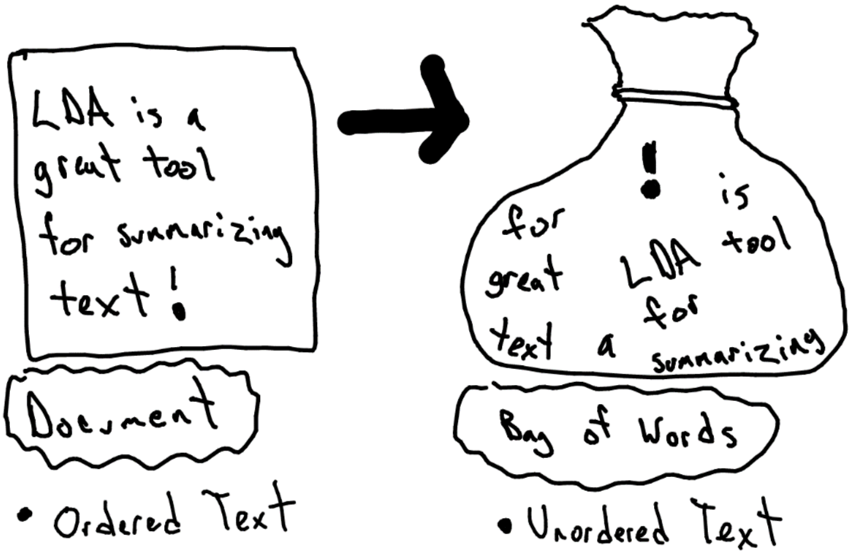
Taxonomy Creation – Predicting Tags that are most likely for any given Question based on Question's text.

It is a Natural Language Processing problem. So we can apply many techniques of NLP freely.

Our options:

- Bag-Of-Words model
- Simple RNN model
- LSTM model
- GRU model

Since in our problem, the questions can be classified based on individual words and there is not so much significance of order in which words appear in the text, we can use the simplest model of all: **Bag-Of-Words model.**


LDA is a great tool for summarizing text!
Document
• Ordered Text
! is for tool great LDA for text a summarizing
Bag of Words
• Unordered Text

High Level Solution Approach: - { Pseudo Code }

Data Exploration

- Check for all NaN values in data.
- Plot tags frequency.
- Check how tags are distributed among question.
- Obtain Conclusion and Strategize next steps.

Data Cleaning and Data Engineering

- Remove unnecessary features from data [Id]
- Drop all rows where Tags column is NaN.
- Select Most Frequent 500 tags
- Create a list of top 500 tags.
- Find indices of examples containing all tags as a subset of top 500 tags.
- Sample 500,000 indices from list of indices obtained from previous step
- Sample the training set using those indices and save.

Further Data Engineering

- Using Regular Expression, clean all the Titles in the *Title* column.
- Separate Code part from the Body and put into *Code* column.

- Clean Body column using Regular Expression
- Similarly, clean Code Column
- Create new Data Frame by adding Title, Body and Code columns, separated by space into a single column.

Tokenize + Remove Stop words + Stemming

- Loop through all the 500,000 examples:
- Tokenize the text
- Remove Stop words from it
- Stem the remaining words
- Join the words again to form a string
- Save the new modified dataset.

Vectorizing

- Apply binary Count Vectorizer on Tags
- Apply Count Vectorizer on Text

Training

- Train Stochastic Gradient Descent model.
- Train Support Vector Classifier
- Train Logistic Regression Classifier

Testing

- Test different models on data.
- Select the best performance model

Models / Algorithms Used:

Bag-Of-Words model

- NLTK Word Tokenizer (*punkt*)
- NLTK Stopwords (English)
- NLTK Snow Ball Stemmer (English)
- Scikit-Learn’s Count Vectorizer

Training

- Scikit-Learn’s OneVsRestClassifier Model (for multi-class multi-label problem)
- Scikit-Learn’s Stochastic Gradient Descent Classifier (SGDClassifier)
- Scikit-Learn’s Support Vector Classifier (SVC)
- Scikit-Learn’s Logistic Regression Classifier (LogisticRegression)

7. Implementation Framework

General Implementation Approach:

The code was implemented on Google Colab Notebooks. Colab runs on Google's powerful Cloud platform and employs Linux based Virtual Machine setup.

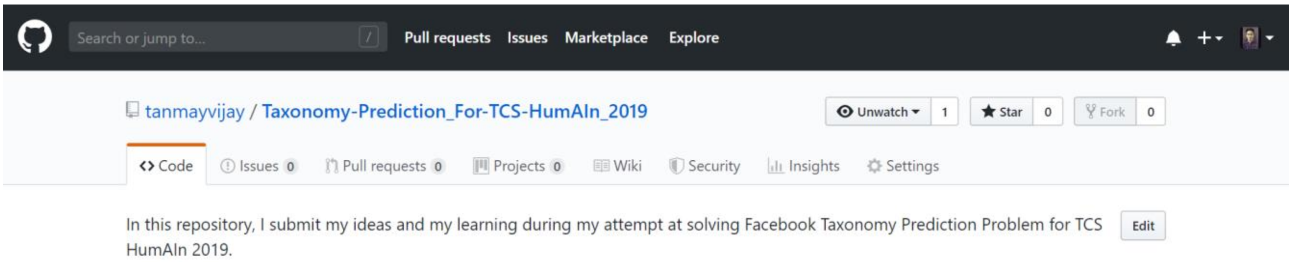
Software Specifications:

- Backend – Jupyter Notebooks
- Language – Python 3.7.4
- Frameworks – Scikit-Learn, Numpy and Pandas

Hardware Specifications:

- CPU – 2vCPU @ 2.2GHz (although lesser will work fine).
- GPU – 1xTesla K80 (although not necessary).
- RAM – 12 GB
- Storage – approx. 5 GB (due to saving data at multiple checkpoints)

8. Solution Submission



In this repository, I submit my ideas and my learning during my attempt at solving Facebook Taxonomy Prediction Problem for TCS HumAIn 2019.

Link to Repository – Taxonomy Prediction Repository

(https://github.com/tanmayvijay/Taxonomy-Prediction_For-TCS-HumAIn_2019)

Link to Main Jupyter Notebook – Taxonomy Prediction Notebook

Instructions:

- Clone the repository.
- Download the dataset from this link.
- Extract the Data into cloned Repository folder.
- Create Virtual Environment and Install requirements specified in "requirements.txt". (Read Appendix for more details)
- Open the "Taxonomy-Prediction-Notebook" in Jupyter Notebook
- Run all cells in the Notebook

9. Appendix

How to install Virtual Environment?

- Open Terminal
- Run the following Command:

Linux or MacOS

```
python3 -m pip install --user virtualenv
```

Windows

```
py -m pip install --user virtualenv
```

How to create a Virtual Environment?

- Open Terminal
- Run the following Command:

Linux or MacOS

```
python3 -m venv env
```

Windows

```
py -m venv env
```

How to activate Virtual Environment?

- Open Terminal
- Run the following Command:

Linux or MacOS

```
source env/bin/activate
```

Windows

```
.\env\Scripts\activate
```

How to install Project requirements?

- Clone the repository
- Open Repository folder
- Open Terminal in the folder. (By pressing shift and right click simultaneously)
- Optionally: Activate the desired virtual environment.
- Run:

```
pip install -r requirements.txt
```

How to setup Google Colab?

- Create a Google account (if you don't have one already).
- Sign In to your Google account.
- Go to: https://colab.research.google.com/ .
- Click on New Python 3 Notebook at the bottom of floating box.

How to clone Github repository?

- Go to: https://github.com/ .
- Search the user or repository you are looking for.
- On the repository page, click **"Clone or Download"**.
- Download Zip or Open on github desktop, if you have the Github Client application installed.
- Unzip the archive in case you downloaded the zip.

What is OneVsRestClassifier?

It is a classifier used to multiclass classification.
One classifier is fitted for each class against all other classes.
For more info: https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

What else can be done to improve Project score?

We can train more robust, more powerful model to boost project score. Models which can do better: Artificial Neural Network, LSTM, GRU, Simple RNN, etc
We can also try training on larger portions of the dataset.

- *All these models may require huge time and computational resources to train.*

One good approach is to use GridSearchCV with all trained models to make them even more robust, but, again that would require huge amounts of time and computational resources.

What is Micro Average F1 score and why is it a good suite for this Problem?

A micro-average f1 will aggregate the contributions of all classes to compute the average metric.
In a multi-class classification setup, micro-average is preferable if you suspect there might be class imbalance (i.e., you may have many more examples of one class than of other classes).

10. References

TCS HumAIn 2019 – For the opportunities and problem set
https://campuscommune.tcs.com/contests/tcs-humain-2019 .

Kaggle for Dataset
https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/rules

Hands On Machine Learning With Scikit Learn And TensorFlow – Book by Aurélien Géron

Google Colaboratory
https://colab.research.google.com/notebooks/welcome.ipynb#recent=true

NLTK (Natural Language Tool Kit)
https://www.nltk.org/

Bag of Words Model
https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Documentation References:

Numpy Documentation
https://www.numpy.org/doc/1.17/

Pandas Documentation
https://pandas.pydata.org/pandas-docs/stable/

Matplotlib Documentation
https://matplotlib.org/#documentation

Scikit Learn Documntation
https://scikit-learn.org/stable/documentation.html

Scipy Documentation
https://www.scipy.org/docs.html

Pickle Documentation
https://docs.python.org/3/library/pickle.html

Blogs that helped a lot:

Machine Learning — Multiclass Classification with Imbalanced Dataset – By Javaid Nabi
https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a

Solving Multi-Label Classification problems (Case studies included) – By Shubham Jain
https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/

Scikit Learn Documnetation
https://scikit-learn.org/stable/auto_examples/plot_multilabel.html

Ways to Solve a MultiLabel Classification Problem – By Jyothish
https://www.kaggle.com/jyothish/ways-to-solve-a-multilabel-classification-problem

Multi Label Text Classification with Scikit-Learn – By Susan Li
https://towardsdatascience.com/multi-label-text-classification-with-scikit-learn-30714b7819c5

11. Final Thoughts

Finally, we reach to the end of my project solution approach.

This project was a great opportunity for me, both in means of learning and creating an actually useful application.

This project is my first real Machine Learning Project. I have put all my efforts and learning into this project and this makes me happy to be better than yesterday. I hope to get more such opportunities from TCS to help me learn and grow while keeping the air of competition.

I feel extremely Thankful to entire TCS community for giving college students chances to learn and explore real world problems, and boosting their confidence through such opportunities.

It would be my pleasure to be a part of this community in coming future. So, its my humble request to all the judges to select my solution for TCS HumAIn 2019 Idea Submission round.

Hope to see you in the next round.

Tanmay Vijay,
CT20192830706
UNIVERSITY COLLEGE OF ENGINEERING, RAJASTHAN
TECHNICAL UNIVERSITY, KOTA