What’s your calling ?

23.12.2018

TEAM BRUH-GRAMMERS,
Contact : Jayasuriya R.S.,
7418261291,
Thiagarajar college of Engineering.

Solution Design :

(Please do read the problem statement from the proposal, beforehand.)

With a sole goal to find the Engineering branch that is suitable for the user, we set out to design “What’s your calling ?”.

We found the importance of ‘curiosity’ and ‘interest’ in creating better Engineers.

Once the user signs up, he/she is shown news articles across various streams in Engineering. The user can click on an article that kindles his curiosity and a page with the article opens up. He/she can read the article and leave a review for our NLP algorithms to take care of. At the bottom of the site, an article related to the one the user has read is also shown. The user can read this or proceed back to the feed. Based on the articles that he clicks and his reviews, the feed learns about the user’s interests and shows content accordingly.

Once, there is enough data about the user, he/she can see the results tab, wherein the top 3 Engineering branches recommended for the user is shown.

Algorithms / Models applied :

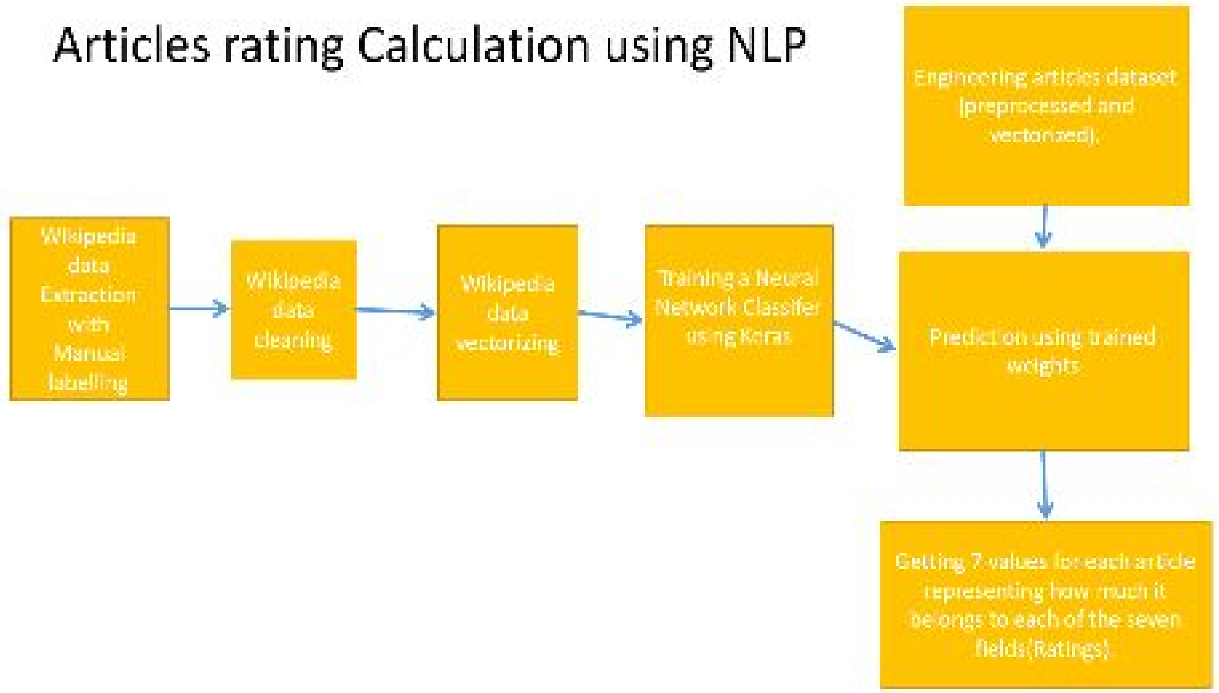
[DETAILED EXPLANATION]

We needed at least a 1000 articles related to Engineering that would later pop up in the user’s news feed. The very idea of manually writing all those articles or doing 1000 * (ctrl-c + ctrl-v) were discarded at the first thought . We decided to scrape content from different websites. We came up with a Node Js script that would do the task for us. The sites mainly scraped were, sciencedaily, allaboutcircuits,geoengineer, etc.,

For more details take a look at https://github.com/jayasuriyars/whatsyourcalling/webscraping

We got a json file with the articles. There were also some redundant data alongside. The json file was then processed and separated to individual articles and they are saved as .pkl files. This was then cleaned and preprocessed.

The thing with Engineering branches are that, they are interdependent with each other. Say, if there is an article about ‘Transistors’, you just can’t classify them as “related to a single branch”. So, we decided to build a model that when provided with an article, will find out how much it belonged to each branch. We chose 7 Engineering branches.



Articles rating Calculation using NLP

Wikipedia data Extraction with Manual labelling
Wikipedia data cleaning
Wikipedia data vectorizing
Training a Neural Network Classifer using Keras
Engineering articles dataset (preprocessed and vectorized).
Prediction using trained weights
Getting 7 values for each article representing how much it belongs to each of the seven fields(Ratings)

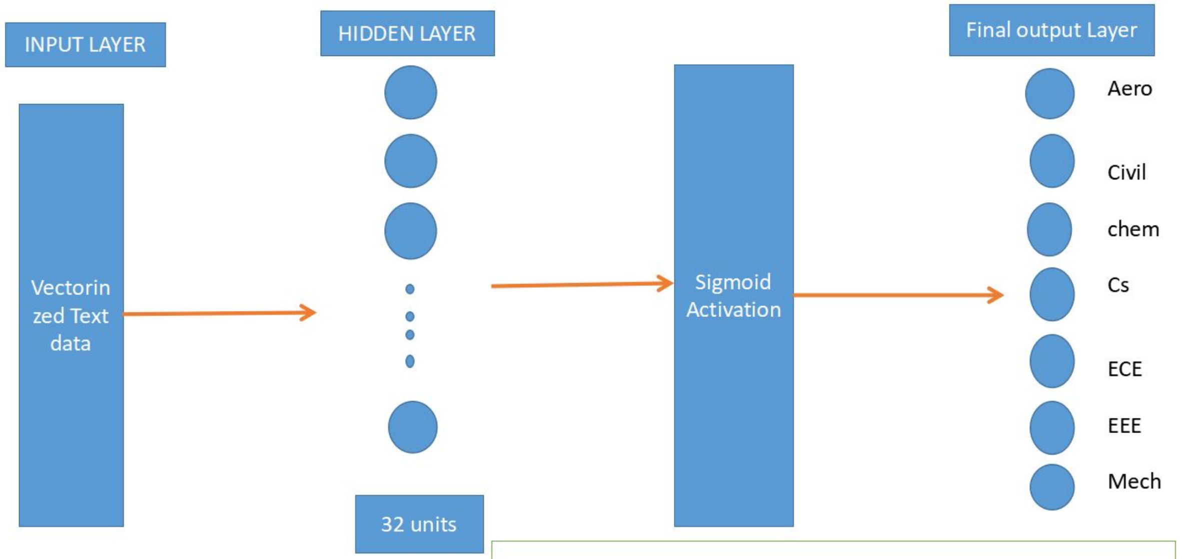
The training data was taken from Wikipedia's pages about each branch using a python script.This data was then cleaned, vectorized (tf_idf) and a neural net classifier was built using keras to train.

The trained weights are stored in a keras model file and the vectorized weights are stored in 'vect.sav' file. The Engineering articles are then vectorized using the 'vect.sav' weights. The vectorized data is then sent through the trained neural net (keras model file). The sigmoid activation function is chosen to calculate the ratings.The rating with 7 values for each articles representing how much it belongs to each Engineering branch are obtained. The results are stored in a .csv file.

These are then used for to select contents to display on the user feed and to calculate the user's affinity towards each branch.

Accuracy obtained - 88%

For more details take a look at https://github.com/jayasuriyars/whatsyourcalling/python



Neural nets Architecture.

The server was built using a Node Js, a relational database (mySql) was chosen to be the DB.

We decided to add two more features, to get to know the user better.

1. The user can review the articles he/she has read.

2. A related post is shown at the bottom for each article he/she has read.

Since, these are real time features and the data are processed dynamically (during the run time), having a model on the python server to do the 'number crunching' (ML part) , a node Js server to render the site and establishing communication between these two, is not very efficient. So, we decided to explore ML with Javascript and have a single node Js server do everything.

The user review system :

The user can provide his review/feedback for each article that he/she has read. This is then processed by another pre-trained NLP model. The words are validated against AFINN dataset. http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

The sentiment of the sentence is obtained and converted to a range of -3 to 3.

The related post system :

For each post, the user has read, there is a related post shown at the bottom of the page. This is done in order inorder to let the user explore the topic he/she has currently read, thereby making the site more engaging to him/her. This is done by another pre-trained model. The current article is chosen and are compared with all the other articles, which the user has not seen yet by the use of tf_idf vectorization, after filtering out the stop words (filler words).

The feed :

Since, we do not have any historic data (other users' data) , the articles have to be recommend only with the current user's data. The articles which the user clicks, the review he/she gives for them are taken into account and the top 3 branches for a user are calculated at any given time by making use of the ratings of articles (already computed using the python model). A score is computed for each article, which the user hasn't yet seen and 5 articles with the highest scores are taken. Also, to avoid monotony of the subjects in the articles, two articles are chosen at random. To eliminate false negatives, the two articles with the least score are also chosen. These constitute the user feed.

Incase of any doubt/clarification, feel free to reach out.

Tech-stack :

I. Web Scraping :

Web Scraping - Node Js

Dependencies :

Cheerio,

Request - promise

II. Articles rating calculation :

Python,

Dependencies :

Sklearn, Keras, Beautifulsoup

III. The site:

HTML5,

CSS,

Javascript,

Node Js,

Express,

MySql,

For the dependencies list, please take a look at the package.json file.

Competitive analysis / Business Impact :

I. First mover advantage :

Not many people, realise the depth of the problem we are trying to solve. For any student enrolled in a branch that he does not like, 8 hours * 22 days * 12 months * 4 years = 8448 hours, is too much time to waste.

"You still can switch over after Engineering."

Easier said than done. Maintaining a good Cgpa in a branch you do not like isn't very easy. Low gpa obviously eliminates placement opportunities in college.(Google says, degree isn't necessary to work at google. Yet has a gpa bar, when it comes to recruiting from Indian colleges.) Staying at other branches and learning Computer Science, maybe a bit easy due to the availability of large no. of. MOOCs and other online resources. But the converse of it is not true. Learning a core subject online, without lab sessions isn't considered very fruitful.

There is no company that operates in this specific space to solve this particular problem. This gives us the first mover advantage.

II. The real competitors ?

Students spend a lot of money to meet "career consulting agencies". They recommend the most hyped up fields based on the CTC offered. Also, their only metric of assessing a student is his/her marks in the board exams / JEE.

We at ‘What’s your calling ?’ assess student’s interests, curiosity and make personalised recommendations for them. Also, they get to read a lot of posts related to Engineering.

This could be a disruptive product and may enrich the quality of Engineering education in India.

III. Business model :

“ Ads are uncool ! ” We are very well aware of this. Just like Wikipedia has Wikimedia foundation, we plan to start a non profit organisation and have people who are benefited by this site, fund us.

Once we get a lot of users, we can have a moderator team. The users who are experts in any Engineering branch can write articles and the moderator team can supervise them.

If this strategy doesn’t work out, we’ll have to add advertisements related only to Engineering. These advertisements could include online courses, electronic equipments, devices like laptops, printers etc.