

Smart Assignment led by AI driven methodologies

Pinniti Gurunaidu

Senior Data Scientist, Insurance Tech Company, Singapore. Email: gurunaidupinninti@gmail.com

Introduction

Often job or task assignments to the work professionals used to be manual in nature or in the semi automated process by using some definite rules. This leads to a problem of finding the right person to attend to the right task. If the complexity of the task increases and finding the suitable professional is even more tedious and time consuming effort. The very same problem can be tackled smartly with the use of Machine learning, Natural Language Processing and Linear Programming methodologies if a certain amount of demographic information of the professionals and transactional details of past jobs or tasks are available in digital records either in flat files or database tables. Let's deep dive into addressing some of the approaches of Smart Assignment.

Before finding a solution to any problem thus identifying methodologies, the following questions need to be well thought of:

- What kind of tasks are going to be handled by workers?
- Do we know much about these workers in terms of their professional expertise ? Is that information readily available ?
- Are we tackling the big problem or small problem ?
- Are we going to throw garbage at the ML models ?
- How big is the task pool data ?
- Are there any required time constraints for making assignments?

Certain degree of understanding on the actual problem statement and purpose of the solution will give us finite clues to tackle the problem in a finite way rather than failing fast.

Although the assignment can be done in various ways, this article is focusing on the one [(1)] of the following two approaches;

(1) Task Driven Approach

Smart Assignment is completely based on the nature of the task. The detailed discussion on this topic covered in the later part of this paper

(2) User Driven Approach

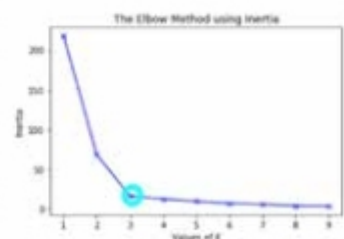
Smart Assignment is completely based on the different users attending the different tasks. In this approach, we try to identify if userA has done task1, can userB be able to attend task2 where task2 is similar to task1? This can be coined as profiling the users based on the tasks they have attended.

Task Driven Approach

This methodology certainly uses all the historical tasks carried out by the workers. Then using clustering and word embedding models, tasks further then be grouped based on the similarities among the tasks. Each group thus contains the similar tasks that were carried out by different or same workers.

Conceptual Understanding

Clustering can be done using Agglomerative Hierarchical clustering[i] or distance based K-means clustering and the word embeddings can be formed using either tf-idf aka Term Frequency - Inverse Document Frequency[ii], word2vec[iii], pre-trained sentence transformer BERT aka Bidirectional Encoder Representations from Transformers models either from spacy or Hugging Face[iiii]. These embeddings will be fed to clustering models and the Elbow method will be used to determine the appropriate number





of clusters[v]. Silhouette Score can be used to further strengthen selecting the right embedding model and usually this score ranges from -1 to 1 where -1 denotes the data points that are not placed in right clusters though the clusters were formed with utmost separable distance between each cluster centroids, 1 denotes the data points that are placed in right clusters that were formed with utmost separable distance between each cluster centroids and 0 denotes that the clusters are completely overlapped or data points are not separable. The higher the positive score of the Silhouette score tends to vote the better the model.



As we are dealing with text based data, it is better to form Word-clouds[vii] using each embedding model and further tune if the model results were not satisfactory. The word clouds needs to be further examined using Domain Knowledge and subject matter expert(SME) advice thus it requires careful pre-processing of the data.

Detailed Methodology

Pre-Processing

Each task that was completed has the username, description, start and end times along with the task notes. Those tasks for certain period data have been collected via a set of SQL queries. For each task, task notes and description were appended in the sequence that were recorded so as not to lose the actual meaning of the task.

Each task further then be named as a document in NLP for understanding purpose. Each document has been pre-processed by removing stop words, special characters and applied lemmatization and stemming (techniques that used to find root and similar words) and using some business oriented regex rules.

Model Training

These processed documents are the final documents that need to be supplied for the embeddings. Pick the standard splitting of the documents as training and test sets. On the training data set, embeddings have been created using different models like tf-idf, pre-trained bert models like all-mpnet-base-v2, paraphrase-MiniLM-L3-v2 etc and those embeddings be further submitted to k-means clustering models to group them into finite number of clusters. Each document will be assigned a cluster number based on k-means fitted model.

Our main task is to identify the best embedding model. This can be achieved in different ways, silhouette score and processing time of the model were used to determine the best model in this paper. The best aka highest positive silhouette score and least processing time that model yields will be voted as the best model. But, strictly depending on this method may induce some incorrect results, so need to apply further business logic by consulting SME(Subject Matter Experts) and further tune the embedding models.

Prediction

On the Test data set, embeddings will be created using the best model that have been selected in the training phase for each document. Then use the same k-means model to predict the cluster for each document. Then validate the clusters manually and with SMEs to further fine tune the models by retraining if the induced results were not satisfactory

Recommendation

For every new task, do the pre-processing, predict the cluster using the chosen embedding model and k-means clustering. Once the cluster is identified, pick the same cluster data from training data set and identify the most frequent unique top n usernames and recommend them as the workers can attend the task to complete the work in the initial stages. Run these recommendations for a certain amount of period and validate the results with SMEs and see if further re-training is needed. If the recommendations were satisfactory for a finite period, proceed to Assignment Phase

Smart Assignment

Once the recommendations were satisfactory, to do the direct worker assignment smartly, there are many ways of doing it. One approach is with the help of Linear Programming[viii]. It does require

- **Initialize Model:** either it could be a maximisation or minimization problem that depends on the motto we are achieving and accordingly that model will be initialised.
- **Define Decision Variable:** The key variables used for optimising the problem are the worker demographics(worker timings, worker experience, education levels, proficiency, past work performance), complexity of the work, etc
- **Define Objective Function:** Pick either maximisation or minimization model
- **Define the Constraints:** for each decision variable need to have boundary conditions and either in combinatory or mono

- **Solve Model:** then solve the problem using the initialised model. The model yields the optimised values for each submitted decision variables

For the new document, predict the cluster and pick the same cluster documents from trained data. Then apply the optimization model on all the available workers from that group, choose the most optimised model score that belongs to the worker and that worker will be assigned to that task to attend the work. On the contrary, the hardest part is to extract real time worker demographics.

Feedback and model Maintenance

Collect the feedback for each assignment, mark 1 if the user accepts the assignment, 0 if not. Get all the feedback score over a period of time and see the drift of the scores over day by day or shift by shift and alert or feedback to the management about the results and also update if the model retraining is required as declining rate is significant compared to acceptance rate or may be some sort of training to be given to workers to understand on the

approach in layman terms.

Future work

The very same assignment problem can be solved using collaborative filtering and different optimization algorithms to further add the smartness into the assignments

References

- [1] <https://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-agglomerative-clustering-1.html>
- [2] https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_832
- [3] https://www.sbert.net/docs/pretrained_models.html
- [4] <https://www.tensorflow.org/tutorials/text/word2vec>
- [5] <https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/>
- [6] <https://towardsdatascience.com/simple-wordcloud-in-python-2ae54a9f58e5>
- [7] <https://machinelearninggeek.com/solving-linear-programming-using-python-pulp/>

About the Authors



Pinniti Gurunaidu is Senior Data Scientist working at Insurance Tech Company, Singapore. He has overall 10+ years experience in IT with 6+ years of DS experience. Leads and develops AI solutions and builds robust Data Pipelines. In leisure times, he writes quotes in Telugu, volunteering and running. He holds Masters in Information Systems From Nanyang Technological University, Singapore.

Kind Attention !

Prospective Contributors of Newsletter Bytes

Forthcoming Issues : **November 2023: IoT, Machines Learning, Cyber Security**

Please note that Cover Theme for the November 2023 issue is **Machine Learning, IoT, Cyber Security**. Please send your contributions by 15th October, 2023.

The articles should be authored as original text. Plagiarism is strictly prohibited.

Please note that Byte is a newsletter for members at large and not a research journal for publishing fullfledged research papers. Therefore, we expect articles written at the level of general audience of varied member categories. Equations and mathematical expressions within articles are not recommended and, if absolutely necessary, should be minimum. Number of pages of the articles should be minimum 3 to 4 pages. Include a brief biography of five to six lines for each author with high resolution author's photograph.

Please send your article in MS-Word format to Chief Editor, Dr. Tirimula Rao Benala in the email ids bytes.newsletter@gmail.com; Issued on the behalf of the Bytes Editorial Board.

Dr. Tirimula Rao Benala
Editor-In-Chief, Bytes - The newsletter