AI based Work Item Triaging

What the future seeks!

Sourav Dutta, Prasun Biswas

Abstract— Smart institutions are looking to integrate and automate their systems for cost-effective growth without impacting client satisfaction. In order to drive client satisfaction, organizations are aligning their solutions in such a way that it caters to client needs. That even means if clients need to raise service requests, they just enter what they want and rest all is taken care by an Artificial Intelligence based engine on a real time basis.

A case in point can be automation of triaging effort of the client service requests and routing them to different teams for further action. Here an automated triage and routing system was developed to perform consistent triage and routing decision, eliminating manual intervention from associates. This algorithm improves quality of service and reduces costs. The developed system consists of Machine Learning Algorithms, Natural Language Processing Algorithms and graphical user interface (GUI) using Microsoft Visual Studio.

Index Terms— Automation, Triage, Routing, Artificial Intelligence, Machine Learning, Natural Language Processing, Manual Effort, Word Cloud, Text Mining

1. Business Problem:

In today's ever changing dynamic world smart institutions are looking to integrate and automate their systems, and streamline operations as much as possible paving path for cost-effective growth while unlocking new opportunities. An integrated platform helps in capturing each and every issues raised by clients and analyzing effort required to serve those requests.

In the business case discussed here, clients raise their requests in the form of general service requests and provide directions in free text form. Those requests are then manually read by associates and sent to relevant teams who can take appropriate action based on client direction provided. With more and more pressure on cost minimization, the question becomes imperative if the associates doing triaging of service requests adding any value to organization?

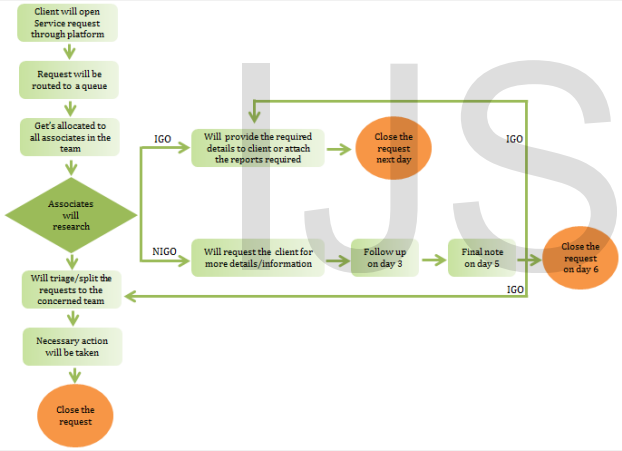

Fig 1: Workflow of Service Request

The team researching the requests can be in multiple forms. There can be one centralized team who does research and then triage the requests to appropriate teams. Now teams receiving the requests may need to classify them further before taking appropriate action. So there can be a small triaging team within each of those teams. In order to make the organization structure lean, the number of such teams should be minimal. An advanced routing system can eliminate such manual intervention from associates and make the process lean.

2. Automatic Triaging Algorithm:

With advancement of Machine Learning techniques and Artificial Intelligence it is possible to build an algorithm that-

i. Reads through instructions present in a request
ii. Understand and classify request type real time

Thus the algorithm becomes intelligent and for future requests raised, it can classify the request type without manual intervention from associates.

Now will go through the methodology and approach in details-

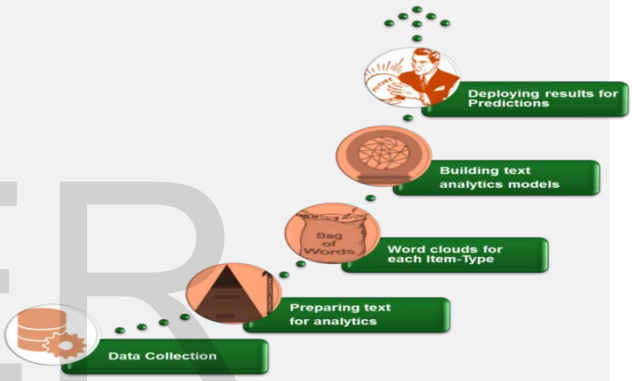

Fig 2: Methodology of Auto Triaging Algorithm

2.1. Data Collection:

Service requests created from Jan'16 to Jul'17 were collected. Each service request comprises of end to end workflow details and client direction.

2.2. Preparing Text Data:

The client directions come in free text form, which are highly unstructured in nature. It becomes imperative to identify the noise in data and eliminate it. Hence the phrases and words which are repetitive in nature and appear in lot of different types of requests are identified. Phrases like "Could you please", "Thank you", "Notification to you" etc. and words like "Request", "Date", "Help", "Need" etc. is not considered as important and omitted.

2.3. Processing Text Data:

Entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing.

Following two steps are mainly followed:

- Noise Removal
- Lexicon Normalization
  - Stemming: rule-based process of stripping the suffixes like 'ing', 'ly', 'es' from word
  - Lemmatization: determines root form of the word, it makes use of vocabulary and morphological analysis (word structure and grammar relations).

In text processing the bag of words approach was followed which does word Lexicon Normalization and the application of a stop word list. In this approach focus is on the words and their statistical distributions. The index tends to be very large, so terms that are grammatically close to each other (like "cell" and "cells") are mapped to one term via word stemming. Also punctuations (like "a", "an", "the") and white spaces are removed from the text data. Thus word clouds are created for each of the teams working on client requests.

Fig 3: Sample Word Cloud

2.4. Feature Engineering:

To analyze the unstructured data, the texts need to be converted into features.

Features can be created using different techniques like:

- Syntactic Parsing like POS tagging
- N-grams
- Topic Modelling using Latent Dirichlet Allocation (LDA)
- Statistical features like TF-IDF
- word2Vec.

Here, the word clouds created for different teams are consolidated under one table along with the frequency of occurrence of each word. The table may become a "Sparse Matrix" depending on number of teams working on client requests.

Team Name	Plan	Source	Employ	Fund	Wire	Forfeiture	...
A	0	0	0	90	0	0	
B	100	0	0	0	0	200	
C	25	0	18	0	0	0	
...							

Fig 4: Word Matrix

For example in Fig 4, the word "Fund" occurs most frequently within client directions part of service requests processed by Team A. Similarly for Team B the words are "Plan" and "Forfeiture" and for Team C "Plan" and "Employ". With more teams added to the tables number of words will also increase. Hence the table may become a sparse matrix which has been the case for the business problem.

The TF-IDF matrix and N-gram combination was used to come up with the feautres. It is numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It converts the text documents into vector models on the basis of occurrence of words in the documents without taking considering the exact ordering.

For Example, let say there is a dataset of N text documents, In any document "D", TF and IDF will be defined as:

Term Frequency (TF) – TF for a term "t" is defined as the count of a term "t" in a document "D".

Inverse Document Frequency (IDF) – IDF for a term is defined as logarithm of ratio of total documents available in the corpus and number of documents containing the term T.

TF-IDF calculates the relative importance of a term in a corpus (list of documents), given by the following formula below:

$$w_{i,j} = tf_{i,j} * \log\left(\frac{N}{df_i}\right)$$

where, $tf_{i,j}$ = # occurance of i in j

df_i = # documents containing i

N = total number of documents

2.5. BEST FIT MODEL SELECTION:

Model development has two parts:

i. Training
ii. Prediction/Testing

Firstly the text input is processes and features are created. The machine learning models then learn from these features and finally is used for predicting against the new text. The model is trained based on one and a half year of data and validted for next few months to track the model performance.

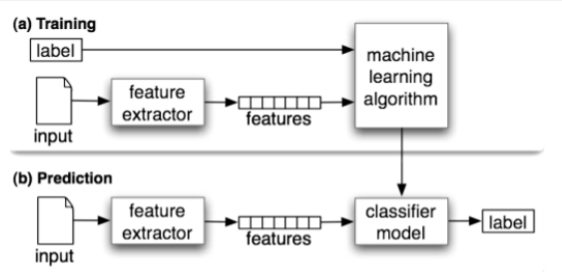

Fig 5: Model Fit Selection*

*Source: https://www.analyticsvidhya.com/

As it is a classic clasification problem, different supervised machine learning techniques like Support Vector Machine (SVM), RandomForest (RF) and Ensemble techniques were tried.

Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes data.

Random Forest use bagging (bootstrapping aggregation) for training/testing. It is used when the variance of a decision tree needs to be reduced. Here idea is to create several subsets of data from training sample chosen randomly with replacement. Now, each collection of subset data is used to train their decision trees. As a result, the output is an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree. Random Forest is an extension over bagging. It takes one extra step where in addition to taking the random subset of data, it also takes the random selection of features rather than using all features to grow trees. When there are many random trees, it's called Random Forest.

Whereas an ensemble contains a number of learners which are usually called base learners. The generalization ability of an ensemble is usually stronger than that of base learners. Actually, ensemble learning is appealing because that it is able to boost weak learners which are slightly better than random guess to strong learners which can make very accurate predictions. Here Basic RF, Extratree Classifier, SGD and Naïve Bayes as base learner were used.

Out of these approaches Random Forest with proper cost function and probability calibration was the best fit for the multiclass problem. Cost function was used to influence the bias in model according to business requirement.

Feature selection was done considering Count matrix(countvectorizer in python), TF-IDF matrix(tfidfVectorizer in python) and LDA based approach.

Metric which has been considered as parameter of best fit model is 'Recall' (recall/sensitivity is the fraction of relevant instances that have been retrieved over the total amount of relevant instances). Higher the 'recall' value with significant accuracy better the model. After multiple iteration and probability calibration:

Recall value of RF : 0.87

Recall value of SVM : 0.81

Recall value of Ensemble learning : 0.83

Techniques	Features	Accuracy Measure
SVM	N-gram	0.72
	LDA	0.76
	TF-IDF	0.81
RF	N-gram	0.79
	LDA	0.84
	TF-IDF	0.87
Ensemble Model	N-gram	0.78
	LDA	0.83
	TF-IDF	0.81

Also cross validation has been performed with 6-fold for each case and the recall for each fold for RF-TF_IDF combination was almost same (~0.86) with precision ~0.83, which ensures model is stable.

2.6. Model deployment for Prediction:

Once the model was built, prediction for next few months were done to check the model performance. For that raw text was passed along with id into the model. It first cleans the text, makes the other modefications and with the defined wordmatrix (TF-IDF), the model predcited the teamname. 'Recall' value was almost same what was obtained from training data.

The biggest advantage of this algorithm is it studies it's patterns and gather more information based on data avaliable and on prior rules together, while running. So it is capable to learn more and predict better in future. However, the false positive predictions can not be fully eliminated. From business stand point its worth to set a cut off on such missclassifications as it might result in client dissatifaction and unwanted finanical penalties. Essentially a right balance needs to be achived between model accuracy and false positive predictions based on business requirement.

3. Conclusion:

Natural Language Processing (NLP) technique has developed in leaps and bounds over the last few years. By combining the power of artificial intelligence, computational linguistics and computer science, Natural Language Processing has helped machines read text by simulating the human ability to understand language. This algorithm is an apt representation of the same.

It definitely eliminates manual triaging work from associates and helps in speeding up the process of request resolution. The benefits can be multifold. These associates can be redeployed within the organization or can be absorbed without backfilling for any attrition. The faster the resolution cycle, the better will be the Client Satisfaction. However, the misclassified service requests routed to inappropriate teams may get delayed in resolution. If those requests are of high priority then Client Satisfaction is likely to be impacted. That is the tradeoff for business. In such scenarios an optimized solution needs to be achieved in consultation with the end stakeholders.

Important Libraries (Python):

[1] Scikit-learn: Machine learning in Python
[2] Natural Language Toolkit (NLTK): The complete toolkit for all NLP techniques.
[3] Gensim: Topic Modelling, word2vec

References:

[1] "Natural Language Processing and Text Mining" by Anne Kao & Stephen R. Poteet
[2] http://www.expertsystem.com/natural-language-processing-applications/
[3] https://www.analyticsvidhya.com/

- *Sourav Dutta, Country: India, PH: +91-9901666223. E-mail: sourav.dutta2010@gmail.com*
- *Prasun Biswas, Country: India, PH: +91-9986839923. E-mail: prasunbiswas90@gmail.com*