Cactus Capstone Project

Document Classifier
Final Report

Submitted By:
Aviral Jain and Uneet Kumar Singh
Students, Postgraduate Diploma in Artificial Intelligence and Machine Learning, Plaksha University

CACTUS

PLAKSHA UNIVERSITY

TABLE OF CONTENTS

Chapter 1: About Project

1.1 Introduction

We (Uneet and Aviral from Plaksha University) did our capstone project with Cactus. The problem statement given to us was to build a rule based/machine learning model that can classify the documents that cactus receives from the researchers. We both worked on this project for two months, April and May 2022. This report captures the approaches and breakthroughs on Document Classifier in a detailed manner.

1.2 Problem Statement

At the time of the capstone, Cactus majorly dealt in 12 document categories which are:

1. **Abstract**
2. Case Report
3. Letter_Email
4. Manuscript for a book
5. Paper for a conference
6. Peer reviewer comments
7. Presentation_Poster
8. Presentation_Speech
9. **Research paper_Journal article**
10. **Response to reviewer comments**
11. Thesis_Dissertation
12. Website content

Given the categories, the problem statement given to us was to build a model that can classify the documents accurately. But the scope was not to build a 12 class classifier model, initially the scope of the project was limited to building a model that can classify the **M**ost **V**aluable **P**apers (MVP) category which are **Abstract**, **Research paper_Journal article (RP)** and **Response to Reviewer Comments (RTRC).**

1.3 Why only MVP categories?

MVP categories witness the maximum traffic (mostly from Japan, China and Korea) and are also responsible for the maximum revenue for Cactus, they form the lion's share of the business that Cactus gets and therefore becomes crucial and a logical starting point.

1.4 Importance of Document Classification

Identifying the type of document is the the starting point of any downstream tasks like editing, peer review, document layout analysis and even billing. The benefits of Document Classification are two pronged; enriching existing tools that we have in Cactus and future proofing:

1.4.1 Immediate benefits

- **Reducing External Dependency**: Currently along with document upload, the client also selects the type of document it is, an in-house document classifier can help in reducing the external dependency and less task for the client
- **Resource Optimization:** Automation will save on a lot of manhours that are currently deployed on checking document type. This is will free up these valuable resources which can be used somewhere more meaningfully
- **Enriching Search Results:** Enriching search results for internal tools like R discovery at the same time provide high-quality input to IDAN and hubble or paperpal.

1.4.2 Future Proofing

- **Dynamic Pricing:** As soon as the client uploads the document, it will be classified. Basis the document type that the client will charged accordingly, making the billing more dynamic and transparent
- **Future Load:** Currently cactus deals with ~5000 docs/day, what will happen once Cactus starts getting 50000 docs/day. Manual intervention will not suffice, We would need a tool that can automate this task, and building and optimisation of this tool will have to start today
- **Competitive Advantage:** Any organization or even countries grow through innovation, this kind of techno;ogical advancements ensure that the company has an edge over competitors and stay relevant for our clients

1.5 How we progressed over 2 months

After having established what the problem statement was and why it is important to solve it (2 prong benefits), let's discuss how we approached this problem.

We progressed towards our goal in 5 stages:

1. **Introduction:** We learnt about Cactus, its business model, its internal tools like IDAN and HUBBLE. We dug deep into our problem statement as to why this problem statement is important and benefits the organization
2. **Dataset Exploration and Rule-Based Approach:** After being convinced about the criticality of the problem, we started going through the documents (a 15 GB corpus of docs was provided to us) and were coming up with simple rules (if else statements) that will help classify the MVP categories
3. **ML classifiers:** We moved up a bit and decided to let machine identify the relevant features that we were doing it manually, we used TF-IDF vectorization, we cleaned the documents first (lowercase, stemming, removing stopwords) and converted them into vectors using TF-IDF vectorizer. We fed these TFIDF into common ML classifiers like Random Forest, XGBoost, SVM etc and noted the accuracy
4. **Deep Learning Models:** We also tried Multi-Layer Perceptron and CNN - 1D, as they are universal approximators and noted the accuracy

5. **Deployment:** Once we had a working model we deployed it on AWS for demo or any integration purpose and also in parallel, built a front end hosted locally using streamlit

We will talk about our progress in detail. We will start with the Dataset Exploration and rule-based modeling.

Chapter 2: Exploratory Data Analysis

2.1 Characteristics of Documents:

1. Abstract: It is a brief summary of a manuscript and with it being placed at the beginning of the manuscript/paper it acts as a point of entry into the paper. Syntactically, it typically has *'Abstract'* as the leading sub-heading with *'keywords'* as the last sub-heading. Abstracts are typically single-paragraph long.
2. Research Paper Journal Article(RP): Research Papers are detailed perusal of a research question. They typically follow a sequential structure with following subjects, inter-alia, addressed in order - *Abstract, Introduction, Methods, Results, Discussion, Acknowledgements, References*. These documents tend to be significantly longer compared to abstracts.
3. Response to reviewer comments(RTRC): These documents are written by researchers/authors in response to comments/feedback received from the reviewers. Style and tone of these documents tends to be apologetic(since authors apologize for crept-in errors) and conversational.

2.2 Feature Analysis:

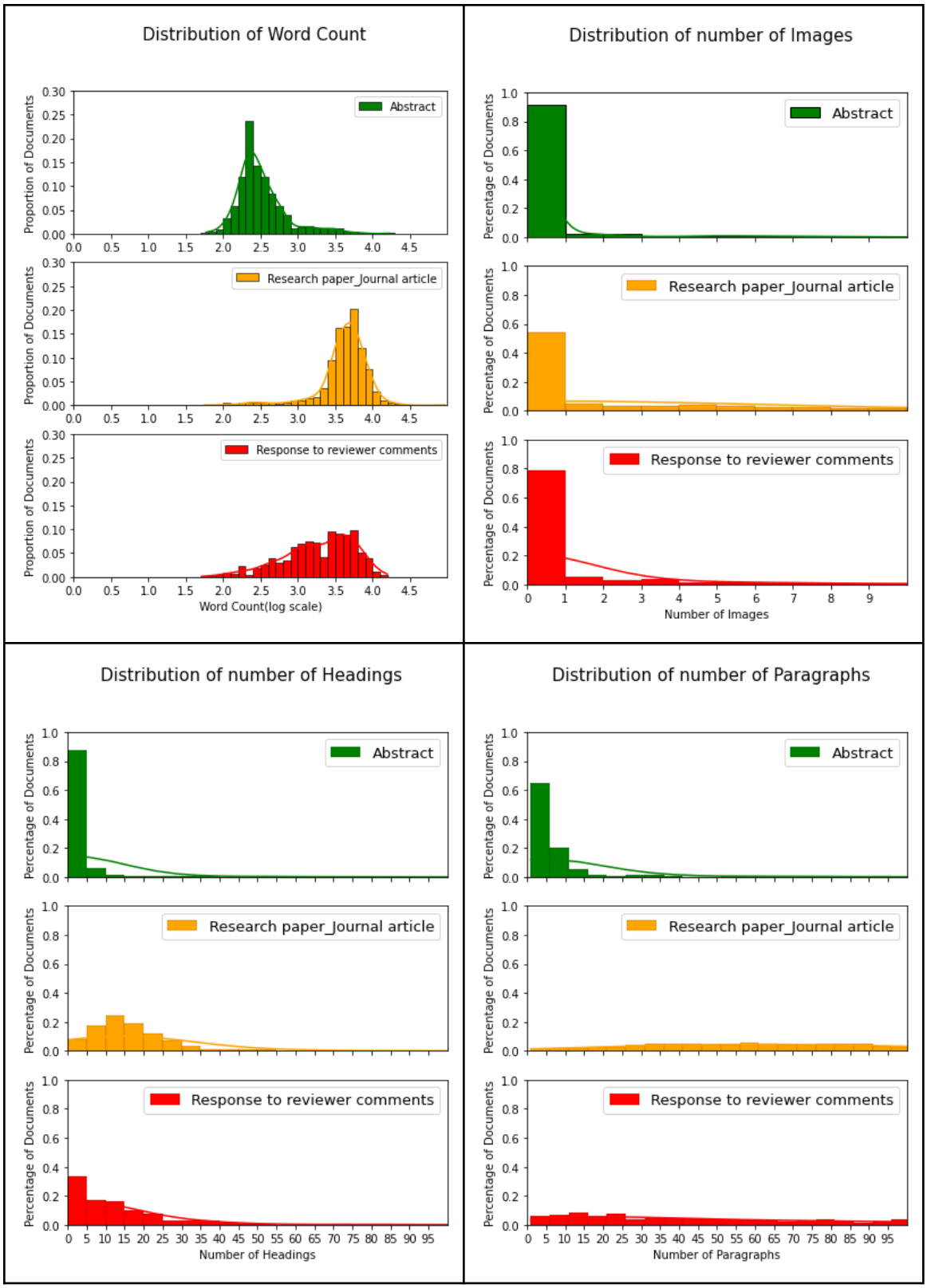
We will analyze every feature and their discriminating power. We will then pick discriminating features and build models on top of those features.

2.2.1 Numerical features:

First, we will analyze the numerical features - *word_count, number of paragraphs, number of images, number of headings.* Refer Figure 2.1 for reference

	Comments
Word Count	Word Count can act as a strong differentiating feature between Abstract and RP. RTRCs have distribution spread all over the range.
Number of Images	There is a very high concentration of documents within 0-5 image range across all three categories. RPs have relatively more images towards the higher side. Number of images therefore have poor discriminating power.
Number of Headings	Abstracts have smaller number of headings and therefore can be segregated on this basis. Distributions of RP and RTRC overlap and therefore cannot be used to discriminate among RP and RTRC.
Number of Paragraphs	Similarly, Abstracts tend to have less number of paragraphs. Distributions of RP and RTRC again overlap and therefore number of paragraphs cannot be used to discriminate among RP and RTRC.

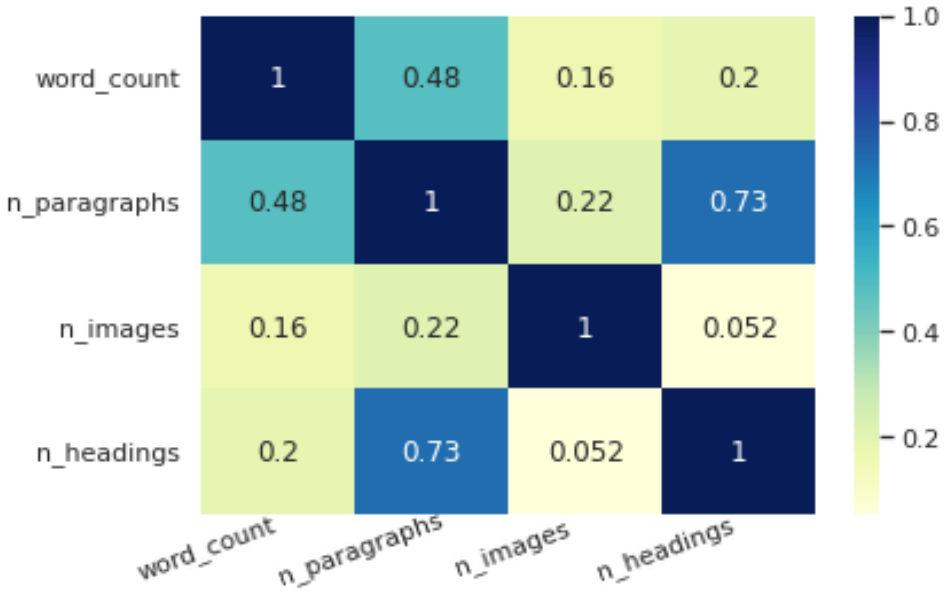
Figure 2.1



We next understand the correlation between these numerical features. Figure 2.1(a) shows correlation coefficient for different feature combinations. *Number of paragraphs, word count* and

number of headings are highly correlated. Distribution of the *number of images* is similar across categories. Therefore we will only consider word count from numerical features.

Figure 2.1(a)



2.2.2 Keyword Analysis(Full Text)

We now analyze what keywords are representative of a document category by considering the entire text/body of a document. A keyword that is appearing across the documents within a category indicates it being representative of that category. If that keyword is found only in that document category and not in others, it becomes an even more strong identifier. We therefore will sort keywords according to their document frequency among documents of that category only. Once we have common keywords for each category, we will explore keywords that are unique to that document category only. As a preprocessing step, we have removed stopwords and have stemmed the keywords to make sure that unnecessary keywords are removed and similar keywords are not counted as different.

2.2.2.1 Abstract:

Distribution of keywords in abstract document types is as shown below in figure 2.2

Figure 2.2

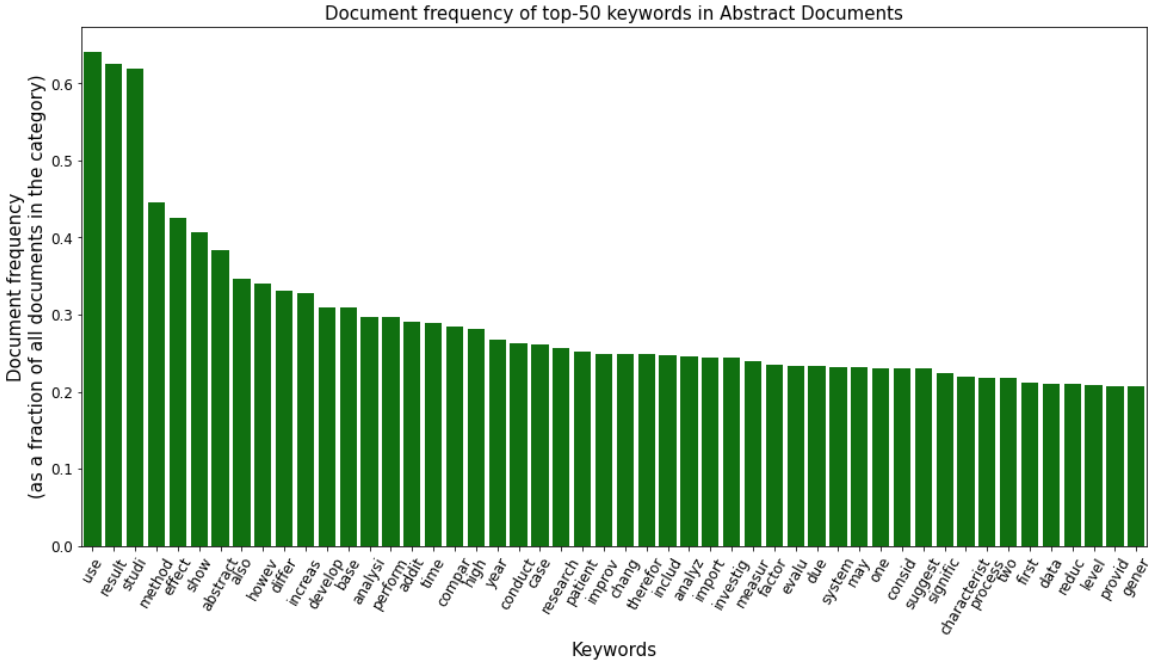
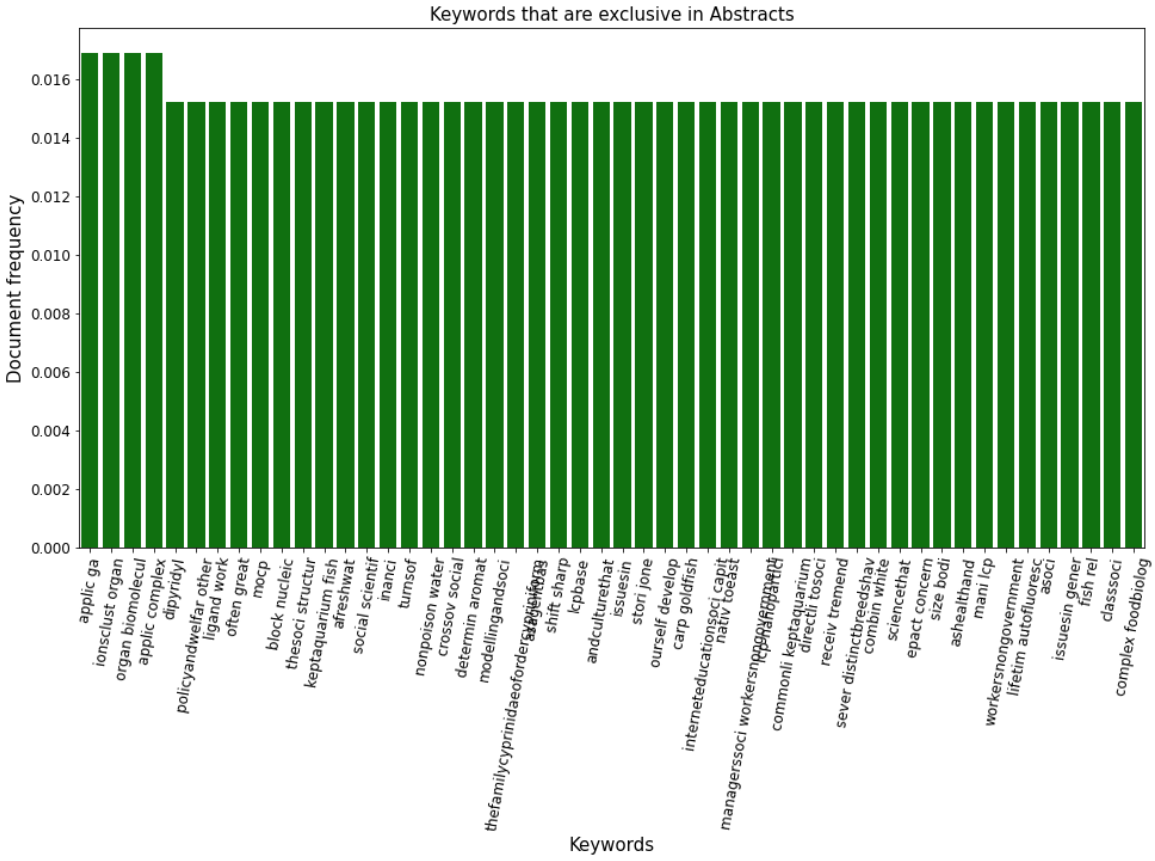
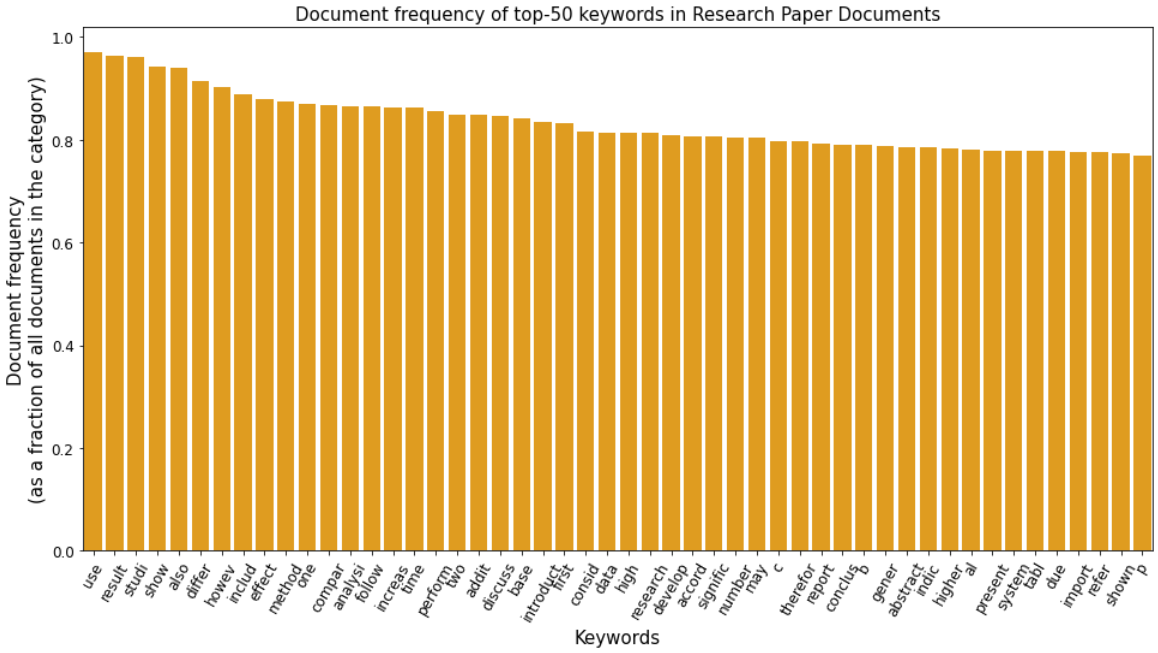

Document frequency of top-50 keywords in Abstract Documents
Document frequency
(as a fraction of all documents in the category)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
use
result
studi
method
effect
show
abstract
also
howev
differ
increas
develop
base
analysi
perform
addit
time
compar
high
year
conduct
case
research
patient
improv
chang
therefor
includ
analyz
import
investig
measur
factor
evalu
due
system
may
one
consid
suggest
signific
characterist
process
two
first
data
reduc
level
provid
gener
Keywords

Figure 2.3


Keywords that are exclusive in Abstracts
Document frequency
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
applic ga
ionsclust organ
organ biomolecul
applic complex
dipyridyl
policyandwelfar other
ligand work
often great
mocp
block nucleic
thesoci structur
keptaquarium fish
afreshwat
social scientif
inanci
turnsof
nonpoison water
crossov social
determin aromat
modellingandsoci
shift sharp
kcpbase
andculturethat
issuesin
stori jone
ourself develop
carp goldfish
internetedu
nativ toeast
commonli keptaquarium
directli tosoci
receiv tremend
sever distinctbreedshav
combin white
sciencethat
epact concern
size bodi
ashealthand
mani icp
workersnongovernment
lifetim autofluoresc
asoci
issuesin gener
fish rel
classsoci
complex foodbiolog
Keywords

Even though ‘abstract’ as a keyword is prominent, there are other keywords that have higher document frequency than this. These other keywords seem generic and can be easily found in other category documents. Given that these words are generic in nature, we will have to explore further if these keywords despite being spread across Abstract documents can act as differentiating features.
We therefore explore keywords that are unique to the Abstract category but common within the Abstract category. Below Figure 2.3 plots these keywords along with their document frequency. Upon perusal, it can be concluded that these keywords are very example-specific keywords and therefore add marginal value.

Figure 2.4

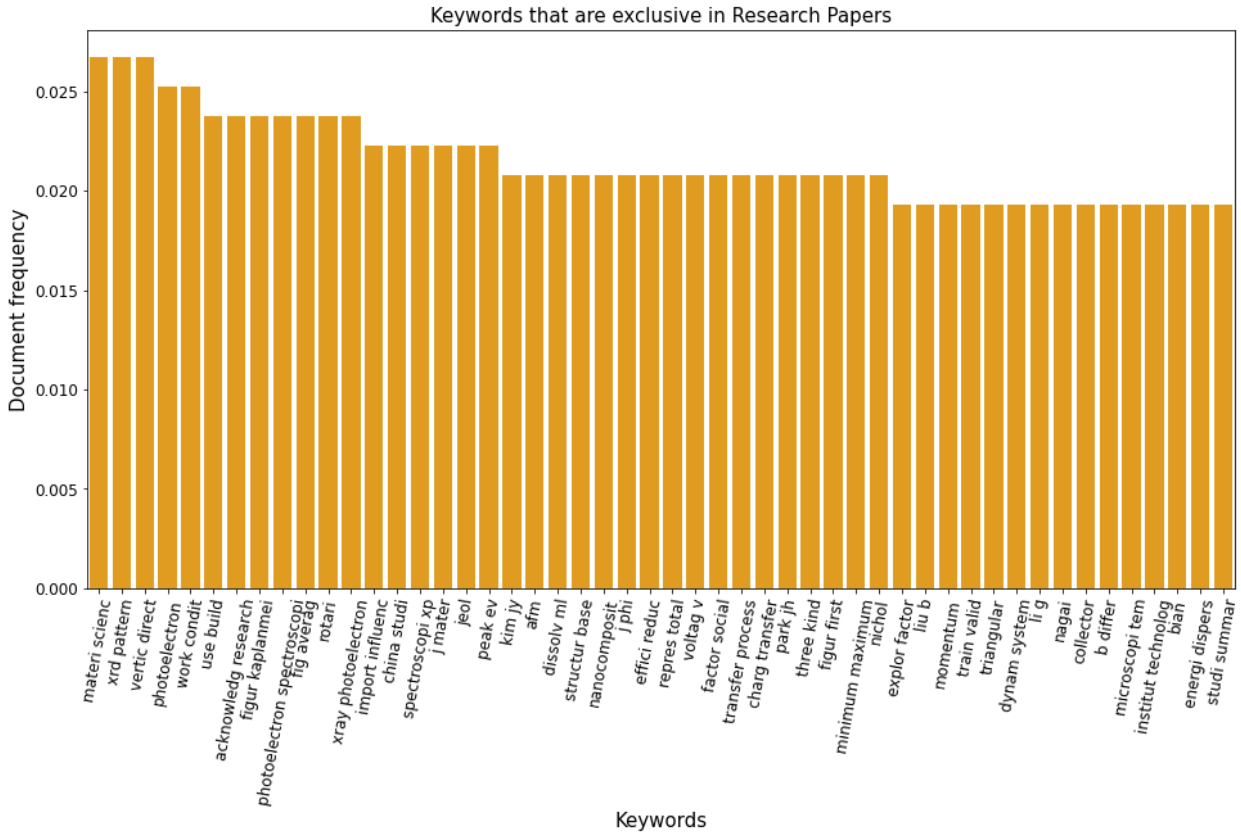


2.2.2.2 Research Paper

Distribution of keywords in Research Paper type documents is as shown in Figure 2.3. It is interesting to observe that there are a lot of common keywords that are found in most of the RP documents. These keywords can act as identifiers but only if they are unique to RP documents. Perusal of these keywords indicate that they are generic keywords and can be found in other document categories also.

This becomes evident when we see common-within-RP-but-unique-to-RP keywords(Figure 2.5). Two things to observe here- First, like Abstract, most of the common keywords among RP are not unique to them; Second, common and unique RP keywords are again example specific keywords.

Figure 2.5



2.2.2.3 Response to Reviewer Comments

Most of the keywords that are common among RTRC documents remain the same. There are few exceptions though with keywords like : *suggest, thank, review, revision*. They have a more prominent incidence in RTRC documents. While these keywords may not be prominent in other categories that need not necessarily mean they are unique.

This is evident from the figure 2.7 that plots keywords common-within-RTRC-but-unique-to-RTRC keywords. Instead of monograms, there are bi-grams like '*thank suggest', 'minor comment'* that are meaningful and are unique only to RTRC documents. This is an encouraging observation and can help in positively discriminating RTRC documents from other documents.

Figure 2.6

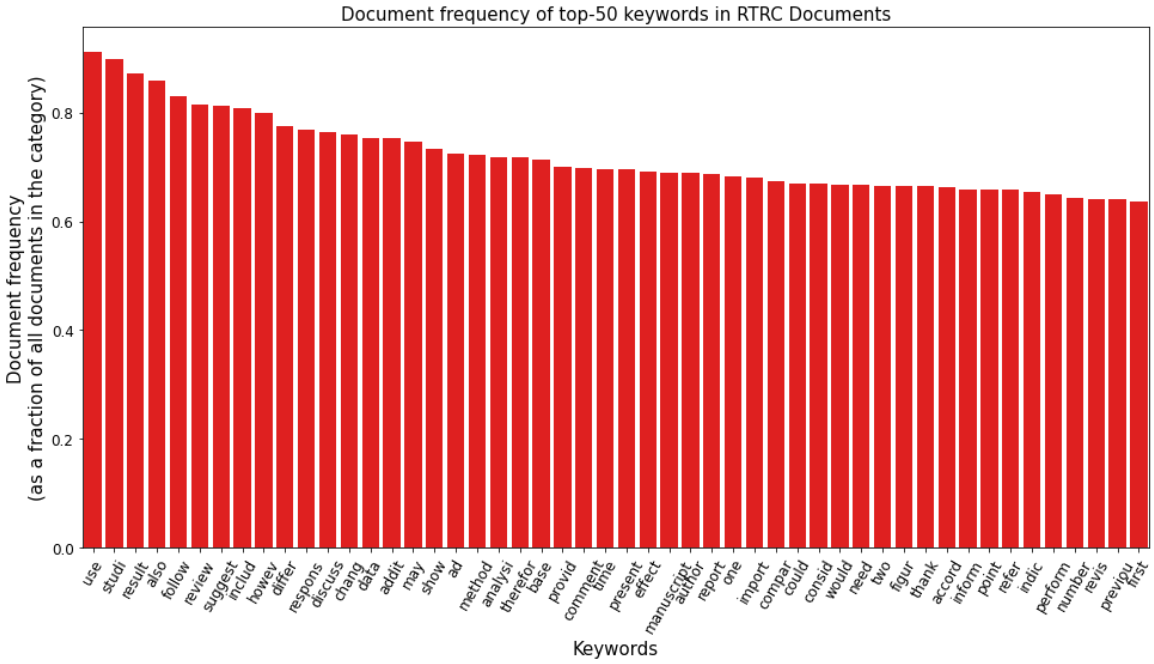
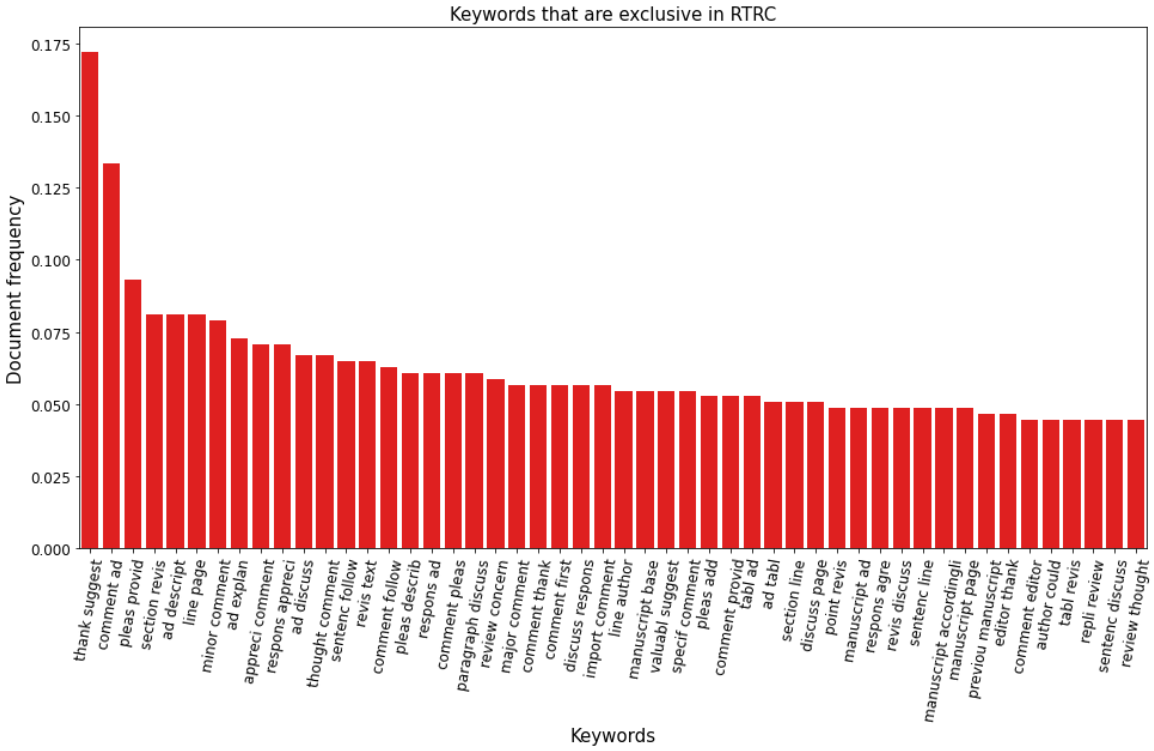

Document frequency of top-50 keywords in RTRC Documents
Document frequency
(as a fraction of all documents in the category)
0.0
0.2
0.4
0.6
0.8
use
studi
result
also
follow
review
suggest
includ
howev
differ
respons
discuss
chang
data
addit
may
show
ad
method
analysi
therefor
base
provid
comment
time
present
effect
manuscript
author
report
one
import
compar
could
consid
would
need
two
figur
thank
accord
inform
point
refer
indic
perform
number
revis
previou
first
Keywords

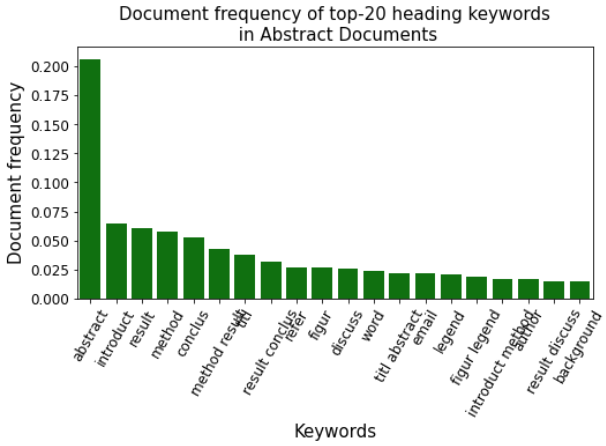
Figure 2.7

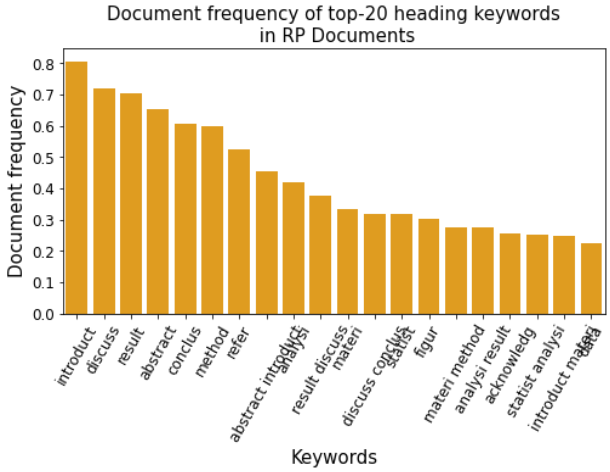

Keywords that are exclusive in RTRC
Document frequency
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
thank suggest
comment ad
pleas provid
section revis
ad descript
line page
minor comment
ad explan
appreci comment
respons appreci
ad discuss
thought comment
sentenc follow
revis text
comment follow
pleas describ
respons ad
comment pleas
paragraph discuss
review concern
major comment
comment thank
comment first
discuss respons
import comment
line author
manuscript base
valuabl suggest
specif comment
pleas add
comment provid
tabl ad
ad tabl
section line
discuss page
point revis
manuscript ad
respons agre
revis discuss
sentenc line
manuscript accordingli
manuscript page
previou manuscript
editor thank
comment editor
author could
tabl revis
repli review
sentenc discuss
review thought
Keywords

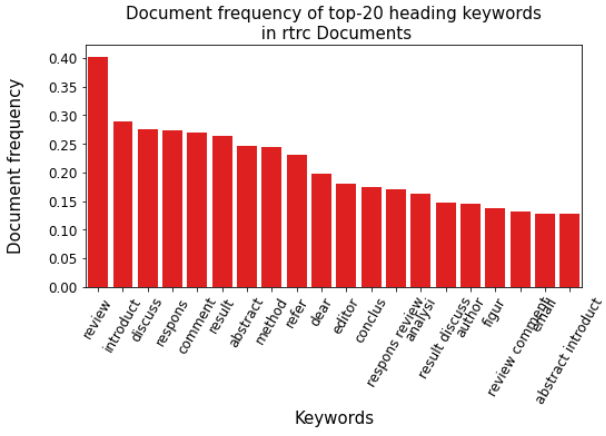
2.2.3 Keyword Analysis(Only Headings)

Here we analyze keywords from headings for all three categories. Document Frequencies for all three document categories are displaced in below figure 2.8

Figure 2.8







Clearly, Abstract and Research Papers share a significant number of headings. This could be because Abstracts are summaries of Research papers and therefore there is a natural overlap between the contents. RTRCs have relatively distinct headings. This is expected since RTRCs are structured differently in terms of layout and syntax.

We tried to find headings that are only found in Abstracts. Results we got were having irrelevant keywords which can be attributed to noise in the document parsing process or those headings being document-specific.
Surprisingly, we didn't get a single heading specific only to Research Paper or only to RTRCs. This could be because of, along with reasons listed above, label noise.

2.3 Conclusion

From the above discussion, following things can be concluded:

1. Numerical features:
    a. Word count can act as a discriminating feature between Abstracts and RPs only.
    b. Other features like number of paragraphs, number of images and number of headings have similar distribution and therefore won't be very useful.
2. Keywords:
    a. There are keywords which are present across the document categories. While there presence alone won't give discriminative power, their relative frequency might provide some discriminating power.
    b. Along with unigrams, bi-grams also have very strong discriminating power in case of RTRCs.
    c. Heading keywords can have positive discriminating power for Research Papers.

Chapter 3: Modeling

3.1 Introduction

In this chapter we will decide evaluation metrics, develop models and then optimize the model. We will explore models from three domains: Rule Based, Deep Learning and Classical Machine Learning.
Below is our document distribution and train_test split description(Figure 3.1):

Figure 3.1 Document distribution in different categories

Dataset		Abstract	Other	Research Paper	Response to Reviewer Comments
Raw DataSet	Train	600	661	667	497
	Test	69	76	73	52
Gold DataSet		18	29	18	20

3.2 Evaluation Criterion

Immediate use case of document classification being built in this project is to segregate documents so that they can be fed into specialized downstream applications. Therefore along with accuracy, we will also be tracking the f1-score and precision(category wise as well as weighted).
We will be doing five-fold cross validation for all models except Rule Based Models. We also want to *caution* about the unstable nature of results on the Gold Dataset. Owing to the small size of it, a single misclassification or correct classification disproportionately manifests into large changes in the evaluation metric. While five fold cross validation is considered as gold standard for evaluation, credibility of cross_validation score is limited by significant label noise in the dataset. Because of the limitations and advantages offered by both the two evaluation types, we will take into account both sets of scores (gold + cross validation)to make a decision.
Because we will also be tracking inference time and vectorisation time, it is important to take note of Hardware specifications. Hardware specification of the machine used are as below:(Figure 3.2)

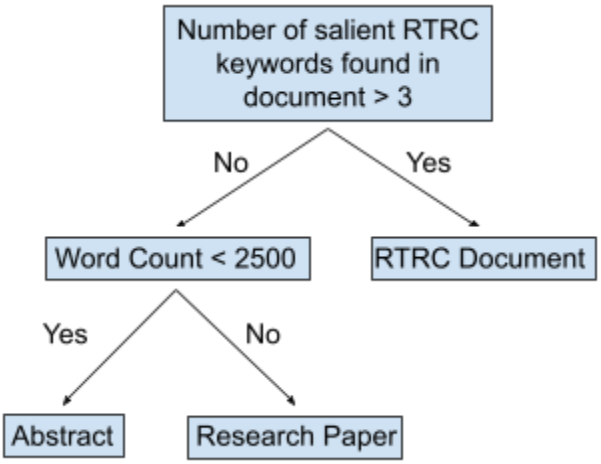
Figure 3.2 : Hardware Specifications

Feature	Detail
Name of Machine	Google Colab Pro Cloud Instance
RAM	24 GB
CPU Model Name	Intel(R) Xeon(R) CPU @ 2.20GHz
Number of CPU Cores	8

3.3 Rule Based Model

Having identified word count as a point of differentiation between Abstract and the rest of the MVP categories, we also needed something that could successfully identify RTRC from a RP. Through keyword frequency analysis and empirical analysis, we identified keywords that are unique to RTRC doc type and they are "`we agree, we appreciate, thank you, reviewer, sincerely, apologize, comment, appreciate your, mentioned, please`".
The presence of any of these words coupled with word count gave us our first and quite a robust model for classification. (Figure 3.3.1)

Figure 3.3.1 : Rules for the Model



Below are the accuracy results of the rule based model on Test dataset(Figure 3.3.2):

Figure 3.3.2

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	Other Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Rule Based Model	Test	0.79	0.83	0.79	-	0.74	1	0.78	-	0.2ms	-
	Gold	0.93	0.94	0.8	-	1	1	0.93	-	0.2ms	-

3.3.1 Limitation of Rule Based Model:

Though rule-based model achieved satisfactory accuracy but there are drawbacks rule based model suffers from:

1. Not Scalable: This is not scalable for more categories. This is because word count threshold for Abstracts and rtrc salient keywords have been arrived at manually with a mix of statistical and empirical approach.

2. Computational speed: Since documents are scanned in linear fashion, reference time increases as length of document increases. On the other hand, if we can use sklearn libraries, faster inferences can be made. This is because of optimisation techniques like vectorised operations, sparse matrix representation.

To overcome these limitations we move over to Machine Learning Models.

3.4 Machine Learning Models:

Before we can apply any Machine Learning model, we will need to vectorise the documents. We have chosen to vectorise documents using TFIDF vectoriser. Also, through GridSearchCV we determined that RandomForestClassifier outperforms other classifiers like DecisionTreeClassifier, SVM Classifier etc. Hereafter, we will explore different experiments and evaluate them on accuracy, speed and memory utilization.

3.4.1 Experiment 1 : Vanilla RandomForest and TF IDF(with and without stemming)

Here we ran a vanilla RandomForest model on the mvp-documents only and then tested it on the mvp testset and gold set. We broke down this experiment into with and without stemming the keywords. Result for the experiment are as follows(Figure 3.4.1):

Figure 3.4.1

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Vanilla Random Forest + Vanilla TFIDF (Only MVP)	Test(MVP)	0.72	0.74	0.71	0.65	0.93	0.71	1.3 ms	1.02 ms	141.3 k
	Gold(MVP)	1	1	1	1	1	1			
Vanilla Random Forest + TFIDF(**with stemming**) (Only MVP)	Test(MVP)	0.72	0.74	0.71	0.65	0.91	0.7	42.5 ms	0.8 ms	119.8 k
	Gold(MVP)	1	1	1	1	1	1			

We get a perfect score in both the cases. In case of stemming, the vectorisation time is 35 times more than what it is for without stemming. We therefore find stemming adding no value to the result.
This model would serve our purpose if there were only three categories to deal with. A model should be able to precisely pick up mvp categories in the presence of Other document types. To generalise for this purpose, we added a fourth category called ‘Other’ to make sure ‘Other’ documents are not classified as one of MVP categories.

3.4.2 Experiment 2 : Vanilla RandomForest and TF IDF(with and without stemming) | MVP + Others

We train the same model as in the previous case but with four document categories. Following are the results(Figure 3.4.2) :

Figure 3.4.2

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Vanilla Random Forest + Vanilla TFIDF (MVP + Others)	Test	0.72	0.74	0.71	0.66	0.93	0.71	1.8 ms	0.9 ms	165.7 k
	Gold	0.82	0.85	0.89	0.63	0.95	0.82			
Vanilla Random Forest + TFIDF(with stemming) (MVP + Others)	Test	0.72	0.74	0.71	0.65	0.91	0.7	57.6 ms	0.8 ms	142.4 k
	Gold	0.75	0.79	0.85	0.6	0.83	0.72			

There are four things to be observed here:

1. Accuracy decreases over previous experiments. This was expected since the 'Other' category is a mix of different document types and thus difficult to extract characteristics for it.
2. Stemming adds no value. Infact, there is a dip in the performance. This could be because stemming takes away the markers of tense and plurality from a keyword which might be having some differentiating capacity.
3. Precision for the Research Papers is particularly low. This could be because there are document types in 'Others' that are very similar to Research Papers.
4. Feature vector length is very large. Even though these vector representations are stored as csr_matrix, it would still be beneficial if this size can be reduced.

We will hereafter focus on increasing the precision for Research paper as well as reduce feature vector length.

3.4.3 Experiment 3 : Reducing Feature Vector Size | Applying min_df filter

From a classification perspective, only those keywords are essential which are repeated in documents of a particular category. Keywords that occur only in a particular document are specific to that document only and hold no representative value. To weed out such keywords, we will put a *min_df* filter in the TFIDF vectoriser. This will make sure only those keywords that are present in the bare minimum of documents are included in the vocabulary for vectorisation. Following are the results of the experiment(Figure 3.4.3(a)):

Figure 3.4.3(a)

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Vanilla Random Forest + TFIDF(Stopword removal + min_df filter) (MVP + Others)	Test	0.73	0.75	0.79	0.67	0.91	0.73	1.6 ms	0.2 ms	0.82k
	Gold	0.85	0.87	0.94	0.69	0.91	0.84			
Vanilla Random Forest + TFIDF(Stopword removal + min_df filter + Stemming) (MVP + Others)	Test	0.74	0.75	0.76	0.67	0.9	0.73	58.2 ms	0.2 ms	0.88k
	Gold	0.86	0.88	0.94	0.72	0.87	0.85			

Following are the observation from the experiment:

1. One can see that performance of the model with respect to previous experiments increases across the metrics. There is a slight slip in the RTRC precision but it can be ignored.
2. Stemming does lead to slightly better performance but that is offset by large vectorisation time.
3. There is a huge drop in the length of the feature vector by a factor of 180 times.

Next we will attempt to Prune the random forest both as a regularizing measure as well as to reduce the size of the model.
Pruning process adopted is inspired from the paper - *Pruning in ordered bagging ensembles*[1]. Steps involved in this pruning process are as follows:

> **Step 1:** Create A signature vector for every estimator(tree) in the ensemble. Signature vector is a one dimensional vector whose length is equal to the number of test examples. And the ith element of this vector is 1 of output of the tree is the same as the ensemble and 0 otherwise.
> **Step 2:** Calculate reference vector for the entire ensemble. This represents the ideal vector where all signatures should have been pointed. It is taken by taking average of all signature vectors.
> **Step 3:** Calculate Orientations(angular distance) of each estimator wrt reference vector using signature vector of each estimator.
> **Step 4:** #Selecting the decision tree that are in the first quadrant of rf_reference_vector and rf_signature_vector i.e. orientation is less than 90 degree.

By taking the above steps we are pruning off decision trees that do not agree with the ensemble outcome and only including those that agree. Below is the performance of the pruned random forest:(figure 3.4.3(b))

Figure 3.4.3(b)

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Random Forest + TFIDF(Stopword removal + min_df filter) (MVP + Others) PRUNING	Test	0.75	0.76	0.92	0.66	0.91	0.74	1.5 ms	0.21 ms	0.82k
	Gold	0.86	0.87	0.94	0.71	0.9	0.86			

Performance improves in terms of the cross_validation scores. On the Goldset, performance remains the same. Till now this is our best model with satisfactory performance on both test(cv scores) and gold dataset. Precision for Research Papers is however still low. We will attempt to address this in the next experiment.

---

[1] *Gonzalo Martínez-Muñoz and Alberto Suárez. 2006. Pruning in ordered bagging ensembles. Proceedings of the 23rd international conference on Machine learning (ICML '06). Association for Computing Machinery, New York, NY, USA, 609–616. https://doi.org/10.1145/1143844.1143921*

3.4.4 Experiment 4 : Forcing model to learn Research Papers | Adding class weights

We increase the weight of the Research Paper class in the RandomForest classifier using the class_weights argument of random Forest. Below are the results for the experiment (Figure 3.4.4):

Figure 3.4.4:

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Random Forest + TFIDF(Stopword removal + min_df filter + **Class_Weight)** (MVP + Others)	Test	0.74	0.75	0.77	0.66	0.92	0.73	1.6 ms	0.18 ms	0.82k
	Gold	0.84	0.86	0.95	0.65	0.9	0.83			
Random Forest + TFIDF(Stopword removal + min_df filter +**Stemming**+ Class_Weight) (MVP + Others)	Test	0.74	0.76	0.74	0.66	0.92	0.73	60.5 ms	0.18 ms	0.88k
	Gold	0.76	0.79	0.85	0.62	0.86	0.74			

Inexplicably, there is no improvement over the previous experiment. In Fact the model performance deteriorates a little. We pruned this model as well but there was no significant improvement. We will therefore attempt to increase the precision of the research papers through a different approach.

3.4.5 Experiment 5 : Forcing model to learn Research Papers | One against the all

In this experiment, we trained a different model for each of the document categories. A Document specific model will be trained to predict whether an input document belongs to that category or not. There were two kinds of decision that we had to make at this stage:

1. Order between different document categories: Since the three models will be operating in a nested manner, we need to decide for what category a document should be tested. Given that precision for RTRC documents has been consistently higher, we will begin with RTRC. At last we will have Research papers since it has the poorest precision.. Flow chart shown in Figure 3.4.5(a) is our flow of decision.
2. Headings for a research paper are a strong marker. So whether to choose just headings or the whole text of a document needs to be decided. We will resolve this question through experiment.

Results of the experiment with headings text for RP Specific model and whole text for RP specific model are as shown in Figure 3.4.5(b).
Following observations can be made:

1. Heading text for the RP model is not leading to good results. Body text based model give better results.
2. Precision for all three mvp categories increases. Precision for RP on the gold set increases to 0.82, highest so far. Cross validation based precision for RP at 0.70 is also highest among all experiments done till now
3. Because there are three models operating in sequence, computational and memory cost is significantly higher. This drawback manifests into insanely high inference time.

Figure 3.4.5(a): FlowChart for Decision Flow

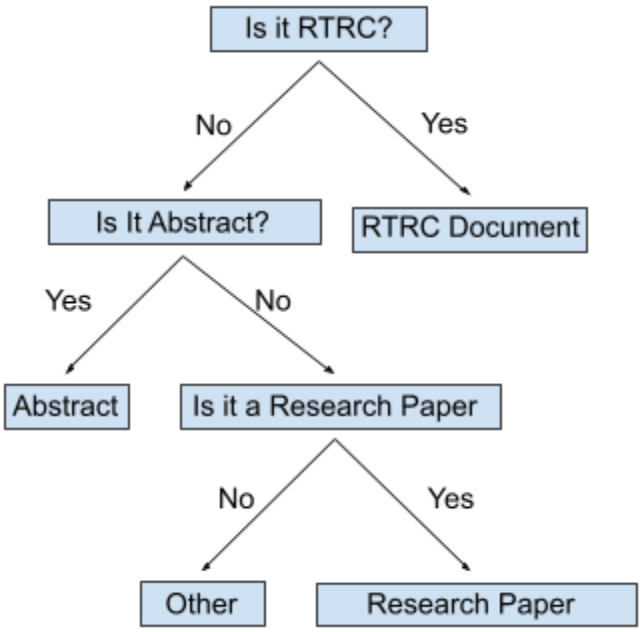


Figure 3.4.5(b): One-against-all experiment results

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
One Against All **Headings Text** for rp-non rp distinction Body text for Abstract and RTRC	Test	0.7	0.73	0.84	0.68	0.94	0.71	7.3 ms	241 ms	-
	Gold	0.8	0.81	1	0.65	0.95	0.8			
One Against All **Body Text** For all three classes	Test	0.67	0.73	0.84	0.7	0.94	0.67	58.3 ms	0.23 ms	0.82k
	Gold	0.85	0.85	0.94	0.82	0.91	0.85			

3.4.6 Experiment 6 : Forcing model to be more precise | From Multiclass to MultiLabel

In this approach we attempt to incorporate the one-against-all paradigm into a single model. To achieve this end, we associate three labels with each label instead of one. Every label will be labeled as whether abstract or not, whether RTRC or not and whether Research Paper or not. An example of how the new labels look is as shown in (Figure 3.4.6(b)).
We conducted this experiment by taking into account our previous learnings i.e. using min_df, removing stopwords, adding class weights. Other than this, we also incorporated bigrams. Results were as shown in Figure 3.4.6(b)

Figure 3.6.1(a): Multi Labeling

	doc_name	doc_category	rp	rtrc	abstract
0	KYOBO_726_input.json	Abstract	non-rp	non-rtrc	abstract
1	ELSVR_68003_input.json	Research paper_Journal article	rp	non-rtrc	non-abstract
2	ELSVR_71326_input.json	Research paper_Journal article	rp	non-rtrc	non-abstract
3	UNCGS_20_input.json	Abstract	non-rp	non-rtrc	abstract
4	HYUGC_20_input.json	Response to reviewer comments	non-rp	rtrc	non-abstract

Figure 3.4.6(b)

Experiment	Dataset	Accuracy	Weighted Precision	Abstract Precision	RP Precision	RTRC Precision	F1-Score weighted	Vectorising time	Inference time(ms)	Feature Vector Length
Random Forest + TFIDF(Stopword removal + min_df filter +Class_Weight) (MVP + Others) **MultiLabel Target**	Test	0.69	0.73	0.8	0.69	0.95	0.69	1.4 ms	0.23 ms	0.82k
	Gold	0.87	0.89	0.94	0.93	1	0.87			
Random Forest + TFIDF(Stopword removal + min_df filter +Class_Weight+Including **Bigrams)** (MVP + Others) **MultiLabel Target**	Test	0.7	0.73	0.81	0.71	0.94	0.71	3.0 ms	0.26 ms	0.83k
	Gold	0.85	0.87	1	0.87	1	0.85			

Following are the observations for this experiment:

1. Precision for Research Paper has increased to 0.93 on Gold dataset. This is the best till now.
2. Overall performance of the model is also best among all till now when tested on Gold Dataset.
3. Adding Bigram helps increase precision of Abstract marginally from 0.94 to 1 (in terms of number of documents, that is one more abstract classified correctly.)

3.5 Conclusion

Out of all the experiments, the pruned version of model produced in Experiment 4.3.3 performs best in cross_validation evaluation while model developed in experiment 3.4.6 performs best on GoldDataset.
For the purpose of deployment, we will use these two models only.

Chapter 4: Error Analysis

4.1 Introduction

To understand why the model was predicting erroneous labels, we sieved the entire pipeline for noise. We found there were three kind of errors/noises:

1. Parsing Error
2. Mislabeling Noise
3. Disambiguation

4.2 Parsing Error

ASPOSE docx parser errs in reading some of the features. Even though it is beyond our scope to address those errors directly, by being aware of error instances we can adjust or replace those features accordingly. Different errors observed in parsing are as follows:

4.2.1 Uppercase text read as Lowercase text:

Text which was in uppercase was parsed and read as a mix of lower and upper case, even though the flag for *all_caps* was correctly marked as *True.* Some examples of such scenarios are listed below:

Erroneous Parsing	Actual Document
all_caps : true text : "Sound barrier at Hiroshima Peace Memorial " style_identifier : 65	**SOUND BARRIER AT HIROSHIMA PEACE MEMORIAL CEREMONY** On August 6 every year, Hiroshima Peace Memorial Ceremony is held in Hiroshima, Japan to com- 2
all_caps : true text : "Abstract"	**ABSTRACT** With the growing deployment of renewable resources worldwide, the Korean government announced the Renewable Energy 3020 Implementation Plan. The goal of this plan is to produce 20% of the country's 3

4.2.2 UpperCase text labeled as Lowercase text:

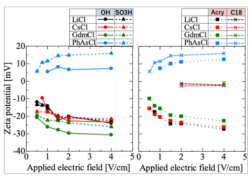
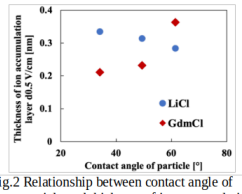
In errors like these, *all_caps* flag in the json file was marked as FALSE. Detailed list of cases where this error occurs can be found in *Attachment_2* in the *Attachment folder.* Some examples of this nature are shown below:

[2] ./Abstract/Physical Sciences/Original/CAOYY_54_input.json
[3] ./Research paper_Journal article/Physical Sciences/Original/CMPQX_4_input.json

Erroneous Parsing	Actual Document
all_caps : false text : "ABSTRACT" all_caps : false text : "PURPOSE"	***ABSTRACT*** PURPOSE Postoperative endophthalmitis is a severe complication after cataract surgery. [4]
all_caps : false text : "MATERIALS AND METHODS"	**MATERIALS AND METHODS** **Patients** Between January 2005 and January 2019, 37 ~~Japanese~~ Asian patients were newly diagnosed [5]

4.2.3 Wrong Image Count:

Number of images were read wrong in some of the cases. Some examples of this nature are shown below:

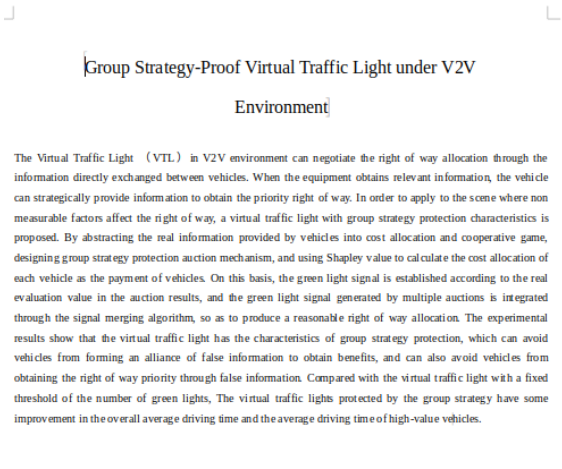
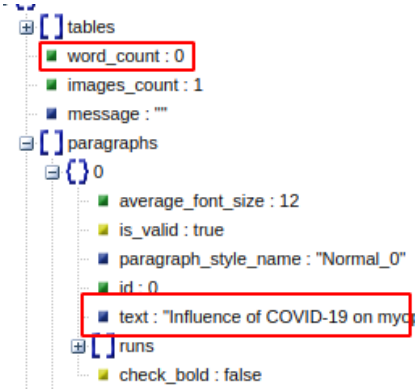
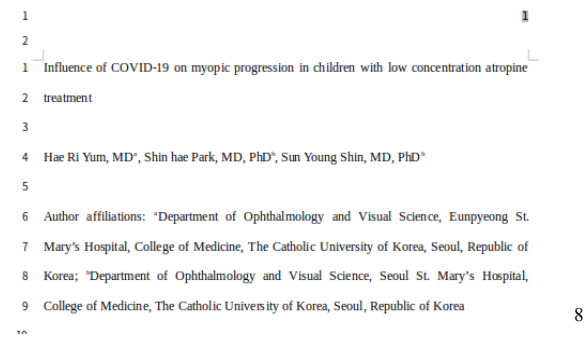
Erroneous Parsing	Actual Document
images_count : 5 message : ""	 Fig. 1 Zeta potential as a function  Fig.2 Relationship between contact angle of [6]

There were also cases where image count was misleading even though not technically incorrect. Special symbols that were placed as inline images were each read as separate images.

4.2.4 Wrong word count:

There were cases where non-null documents were labelled as documents with word_count as zero. Detailed list of cases where this error occurs can be found in *Attachment_3* in the *Attachment folder*. Some of the cases are shown below:

[4] ./Research paper_Journal article/Medicine/Original/JCETA_4_input.json
[5] ./Research paper_Journal article/Medicine/Original/FLMUA_4_input.json
[6] ./Abstract/Physical Sciences/Original/VRDGW_27_input.json

Erroneous Parsing	Actual Document
	
	

4.3 Mislabeling Noise

There were documents whose categories were misclassified. Such mislabelling makes it difficult for a model to learn category specific features. While we knew that there is mislabelling, we wanted to understand the scale of mislabelling. Towards this goal, along with an SME we manually labeled 99 randomly picked documents. We then compared these labels/categories with the labels available from Editage CRM. Below are the results:

[7] ./Abstract/Physical Sciences/Original/COMPE_50_input.json
[8] ./Research paper_Journal article/Medicine/Original/MHYUM_1_input.json

Table 4.1 Scale of Mislabeling noise

	Percentage of Documents
Author and SME Label Matched	82.83%
Author, SME and CRM Label Matched	46.46%
SME and CRM Label Matched	57.58%
Author and CRM Label Matched	52.53%

4.4 Disambiguation

There are categories of documents that are syntactically and layout-wise very similar. Mislabeling noise is therefore not just a result of error in assigning the correct category but also because there are documents whose document category is ambiguous to begin with. For eg: Many of the Conference papers look exactly similar to Research Papers. Response to Reviewer comments and Reviewer comments have similar sentence formation and layout.

4.5 Conclusion

Error in the ASPOSE parser output cannot be dealt with at this stage. To sidestep these errors, we calculated word count separately and didn't rely on the word count key/flag of the JSON file. Label Noise is the most damaging noise for the models. It would be prudent to get documents labeled manually by SMEs and then use them for training purposes. To address the problem of disambiguation, similar documents can be clubbed together into a broader bucket whose distinctive features are easier to extract.