LINGUISTIC ANALYSIS OF RESEARCH PUBLICATIONS ABSTRACTS

A PREPRINT

Aman Kumar
Department of Mathematics
IISER Bhopal
amank17@iiserb.ac.in

Prithvi Poddar
Department of Electrical Engg and Computer Science
IISER Bhopal
prithvid17@iiserb.ac.in

December 6, 2020

GitLab code[1]

ABSTRACT

***K*eywords** Linguistic · POS · Ambiguity · Multilabel Text Classification · NLTK

1 Introduction

In this project, we propose to conduct a linguistic analysis on the abstracts of research publications from the following six disciplinary (or class)

1. Computer Science
2. Physics
3. Mathematics
4. Statistics
5. Quantitative Biology
6. Quantitative Finance

The aim of this study will be to identify the linguistic features and patterns that will be the best choice as parameters, for classifying the publications, into the classes as mentioned above. After having completed the analysis, if time permits, we can test our study by running a classifying on the parameters of our finding and use our collected data about these parameters, to examine how well our classifying performs in classifying unseen publications.

2 Approach

In this section we will build corpus of each class and vocabulary of each discipline.

2.1 Corpus Building

We will use the dataset provided here: Janatahack-Independence Day 2020 ML Hackathon to create corpus of each class . This dataset consists of a list of 20972 research publication titles and their abstracts, along with a tag of which class they belong to. So from this dataset we segregate corpus for each class [title + abstract] in a csv file. Python code for doing this in ***segregate.py***

[1] https://gitlab.com/prithvi-poddar/linguistics_final_project

2.2 Vocabulary

From the above created corpus we extracted the vocabulary for each class [title + abstract] and stored into data folder in csv format. Python code for doing this in ***getvocabdata.py***

2.3 Type Token Ratio

The Type-Token Ratio (TTR), a measure of lexical diversity, which is the ratio obtained by dividing the types (the total number of different words) occurring in a text or utterance by its tokens (the total number of words). A high TTR indicates a high degree of lexical variation while a low TTR indicates the opposite.

we have calculated TTR ratio for each class [title + abstract] and stored into data folder in csv format. Python code for doing this in ***getvocabdata.py***

Table 1: Type Token Ratio.

Computer Science	**Physics**	**Mathematics**	**Statistics**	**Quantitative Biology**	**Quantitative Finance**
Title , Abstract	Title , Abstract	Title , Abstract	Title , Abstract	Title , Abstract	Title , Abstract
6.389 , 39.339	5.411 , 30.350	5.967 , 29.887	6.066 , 34.905	2.239 , 9.872	2.038 , 6.958

Remark The closer the TTR is to 1 the more lexical variety there is.

- Quantitative Biology and Quantitative Finance has more lexical variety both in title + abstract.
- Computer Science , Physics , mathematics , statistics have relatively less lexical variety both in title + abstract.
- Differences in these two lexical variety act as important parameter for classification.

3 POS tagging

POS Tagging simply means labeling words with their appropriate Part-Of-Speech. POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get pos tags and it works after tokenization process.

Method we use in tagging our corpus

- **Universal tagger**
- **Stanford tagger**

Figure 1: POS appendix

- ADJ: adjective
- ADP: adposition
- ADV: adverb
- AUX: auxiliary
- CCONJ: coordinating conjunction
- DET: determiner
- INTJ: interjection
- NOUN: noun
- NUM: numeral
- PART: particle
- PRON: pronoun
- PROPN: proper noun
- PUNCT: punctuation
- SCONJ: subordinating conjunction
- SYM: symbol
- VERB: verb
- X: other

we have calculated POS tag sequence for each Corpus (class) and stored into data folder in csv format. Python code for doing this in ***pos tagging.py***

we have implemented universal taggers for each Corpus (class) and stored into data folder in csv format. Python code for doing this in ***universal tagging.py***

we have analyse Ngram POS tag sequence for each Corpus (class) and stored into data folder in csv format. Python code for doing this in ***ngram analysis.py***

3.1 N-gram tag sequence

Stochastic POS tagger: If a word is tagged with a specific tag in training sentence, analyzing the highest frequency or probability, that word will be given a special tag. This is also called an n-gram approach referring to the fact that the word is decided based on the probability with n previous tags.

we have calculated Ngram POS tag sequence for each Corpus (class) and stored into data folder in csv format. Python code for doing this in ***tagsequences.py***

Figure 2: bi-gram POS taging

Computer Science		Physics		Mathematics		Statistics		Quantitative Biology		Quantitative Finance	
bigram	count	bigram	count	bigram	count	bigram	count	bigram	count	bigram	count
('JJ', 'NN')	0.050211124	('JJ', 'NN')	0.053612223	('IN', 'DT')	0.048912547	('JJ', 'NN')	0.052726496	('JJ', 'NN')	0.053862206	('JJ', 'NN')	0.053239437
('NN', 'IN'	0.044105069	('IN', 'DT')	0.051187894	('JJ', 'NN')	0.045663147	('NN', 'IN')	0.044325039	('NN', 'IN')	0.047413793	('NN', 'IN')	0.051498079
('DT', 'NN'	0.043867129	('NN', 'IN')	0.047440071	('NN', 'IN')	0.044136743	('DT', 'NN')	0.04219144	('IN', 'DT')	0.041897578	('DT', 'NN'	0.05021767
('IN', 'DT')	0.041420849	('DT', 'NN'	0.041185089	('DT', 'NN'	0.041192874	('IN', 'DT')	0.042150032	('DT', 'NN'	0.040084496	('IN', 'DT')	0.048143406
('DT', 'JJ')	0.032573983	('DT', 'JJ')	0.038183624	('DT', 'JJ')	0.037006846	('DT', 'JJ')	0.034252009	('JJ', 'NNS'	0.034508415	('DT', 'JJ')	0.037490397
('NN', 'NN	0.030409373	('NN', 'NN'	0.029792881	('JJ', 'NNS'	0.02392535	('NN', 'NN')	0.02964047	('DT', 'JJ')	0.032857827	('NN', 'NN'	0.034110115
('JJ', 'NNS'	0.027174675	('JJ', 'NNS'	0.027091562	('NNS', 'IN'	0.0229186	('JJ', 'NNS')	0.029013902	('IN', 'JJ')	0.027213328	('JJ', 'NNS'	0.025761844
('NNS', 'IN	0.021977937	('NNS', 'IN'	0.023200346	('NN', 'NN'	0.02103783	('IN', 'JJ')	0.023130703	('NN', 'NN'	0.026913998	('NNS', 'IN'	0.024046095
('IN', 'JJ')	0.021882761	('IN', 'JJ')	0.022663288	('IN', 'JJ')	0.019808191	('NNS', 'IN')	0.022520481	('NNS', 'IN'	0.025271962	('IN', 'JJ')	0.021485275
('NN', '.')	0.016891166	('NN', '.')	0.015583571	('NN', '.')	0.013673775	('NN', 'NNS')	0.016880281	('IN', 'NN')	0.018772236	('NN', 'NN	0.019539052

Remark : IN , DT is most common bi gram pos tag in mathematics , where as JJ , NN is most common bi gram pos tag in others.

hence, act as important parameter for classification.

Figure 3: Tri-gram POS taging

Computer Science		Physics		Mathematics		Statistics		Quantitative Biology		Quantitative Finance	
trigram	count	trigram	count	trigram	count	trigram	count	trigram	count	trigram	count
('DT', 'JJ', 'NN')	0.022293061	('DT', 'JJ', 'NN')	0.025048451	('DT', 'JJ', 'NN')	0.023289273	('DT', 'JJ', 'NN')	0.023135087	('DT', 'JJ', 'NN')	0.022364381	('DT', 'JJ', 'NN')	0.025890548
('IN', 'DT', 'NN')	0.01936897	('IN', 'DT', 'NN')	0.02050528	('IN', 'DT', 'NN')	0.020454341	('IN', 'DT', 'NN')	0.019055309	('IN', 'DT', 'NN')	0.019080281	('IN', 'DT', 'NN')	0.02294553
('DT', 'NN', 'IN')	0.016076393	('NN', 'IN', 'DT')	0.020016317	('NN', 'IN', 'DT')	0.018246749	('DT', 'NN', 'IN')	0.015803691	('DT', 'NN', 'IN')	0.017275736	('NN', 'IN', 'DT')	0.020333427
('NN', 'IN', 'DT')	0.014674475	('IN', 'DT', 'JJ')	0.018538739	('IN', 'DT', 'JJ')	0.017492938	('NN', 'IN', 'DT')	0.015295898	('NN', 'IN', 'DT')	0.01553961	('DT', 'NN', 'IN')	0.019309073
('IN', 'DT', 'JJ')	0.013129794	('DT', 'NN', 'IN')	0.016250677	('DT', 'NN', 'IN')	0.017025875	('IN', 'DT', 'JJ')	0.014017699	('JJ', 'NN', 'IN')	0.014906737	('IN', 'DT', 'JJ')	0.015877487
('JJ', 'NN', 'IN')	0.01291822	('JJ', 'NN', 'IN')	0.014985073	('JJ', 'NN', 'IN')	0.013775219	('JJ', 'NN', 'IN')	0.013849887	('IN', 'DT', 'JJ')	0.014222549	('JJ', 'NN', 'IN')	0.015416528
('JJ', 'NN', 'NN')	0.00969381	('JJ', 'NN', 'NN')	0.009749872	('JJ', 'NNS', 'IN')	0.008836626	('JJ', 'NN', 'NN')	0.010125346	('NN', 'IN', 'JJ')	0.011195019	('JJ', 'NN', 'NN')	0.010883761
('NN', 'IN', 'JJ')	0.009013429	('NN', 'IN', 'JJ')	0.009378474	('NNS', 'IN', 'DT')	0.008636277	('NN', 'IN', 'JJ')	0.009233985	('IN', 'JJ', 'NNS')	0.01087003	('DT', 'NN', 'NN')	0.00957771
('IN', 'JJ', 'NNS')	0.00829768	('JJ', 'NNS', 'IN')	0.008932261	('NN', 'IN', 'JJ')	0.008141666	('IN', 'JJ', 'NNS')	0.00869786	('JJ', 'NNS', 'IN')	0.010151633	('NN', 'IN', 'JJ')	0.009167968
('JJ', 'NNS', 'IN')	0.007954917	('NNS', 'IN', 'DT')	0.008931371	('IN', 'JJ', 'NNS')	0.007427925	('JJ', 'NNS', 'IN')	0.008437425	('IN', 'JJ', 'NN')	0.00951876	('NNS', 'IN', 'DT')	0.008937489

Remark : DT , JJ , NN is most common Tri gram pos tag in all classes ,also IN , DT , NN is second most common Tri gram pos tag in all classes.

hence, can't act as important parameter for classification.

Figure 4: Quad-gram POS taging

Computer Science		Physics		Mathematics		Statistics		Quantitative Biology		Quantitative Finance	
4gram	count	4gram	count	4gram	count	4gram	count	4gram	count	4gram	count
('IN', 'DT', 'JJ', 'NN')	0.008851379	('IN', 'DT', 'JJ', 'NN')	0.012062883	('IN', 'DT', 'JJ', 'NN')	0.010838873	('IN', 'DT', 'JJ', 'NN')	0.009333156	('IN', 'DT', 'JJ', 'NN')	0.009937909	('IN', 'DT', 'JJ', 'NN')	0.010832821
('DT', 'JJ', 'NN', 'IN')	0.007789007	('DT', 'JJ', 'NN', 'IN')	0.008918019	('DT', 'JJ', 'NN', 'IN')	0.008944324	('DT', 'JJ', 'NN', 'IN')	0.008200973	('DT', 'JJ', 'NN', 'IN')	0.008740571	('DT', 'JJ', 'NN', 'IN')	0.009808441
('NN', 'IN', 'DT', 'NN')	0.006694482	('NN', 'IN', 'DT', 'NN	0.008071907	('IN', 'DT', 'NN', 'IN')	0.007226333	('NN', 'IN', 'DT', 'NN')	0.006744063	('NN', 'IN', 'DT', 'NN')	0.007030087	('NN', 'IN', 'DT', 'NN')	0.009706003
('IN', 'DT', 'NN', 'IN')	0.005752367	('NN', 'IN', 'DT', 'JJ')	0.007261421	('NN', 'IN', 'DT', 'NN')	0.007200037	('IN', 'DT', 'NN', 'IN')	0.005895198	('IN', 'DT', 'NN', 'IN')	0.006876144	('IN', 'DT', 'NN', 'IN')	0.007196271
('DT', 'NN', 'IN', 'DT')	0.004976167	('IN', 'DT', 'NN', 'IN')	0.006893585	('NN', 'IN', 'DT', 'JJ')	0.006843166	('NN', 'IN', 'DT', 'JJ')	0.005153123	('NN', 'IN', 'DT', 'JJ')	0.005379471	('DT', 'NN', 'IN', 'DT')	0.007017005
('NN', 'IN', 'DT', 'JJ')	0.004677777	('JJ', 'NN', 'IN', 'DT')	0.006445591	('DT', 'NN', 'IN', 'DT')	0.006424937	('DT', 'NN', 'IN', 'DT')	0.005003836	('DT', 'NN', 'IN', 'DT')	0.005208422	('NN', 'IN', 'DT', 'JJ')	0.006786519
('DT', 'JJ', 'NN', 'NN')	0.004588389	('DT', 'NN', 'IN', 'DT'	0.006407293	('JJ', 'NN', 'IN', 'DT')	0.005986674	('JJ', 'NN', 'IN', 'DT')	0.004858908	('JJ', 'NN', 'IN', 'DT')	0.004729487	('JJ', 'NN', 'IN', 'DT')	0.006197501
('JJ', 'NN', 'IN', 'DT')	0.004287427	('DT', 'JJ', 'NN', 'NN'	0.005118531	('DT', 'JJ', 'NN', 'NN')	0.004006977	('DT', 'JJ', 'NN', 'NN')	0.004749939	('NN', 'IN', 'JJ', 'NNS')	0.004575544	('DT', 'JJ', 'NN', 'NN')	0.005890186
('IN', 'DT', 'NN', 'NN')	0.003506082	('IN', 'DT', 'NN', 'NN	0.004657178	('DT', 'NN', 'IN', 'JJ')	0.003389652	('NN', 'IN', 'JJ', 'NNS')	0.003518594	('DT', 'NN', 'IN', 'JJ')	0.004404495	('IN', 'DT', 'NN', 'NN')	0.005045073
('NN', 'IN', 'JJ', 'NNS')	0.003448848	('DT', 'NN', 'IN', 'JJ')	0.003569701	('DT', 'JJ', 'JJ', 'NN')	0.003344574	('IN', 'DT', 'NN', 'NN')	0.003495711	('NN', 'IN', 'JJ', 'NN')	0.004096608	('NN', 'IN', 'JJ', 'NN')	0.003969473

Remark : IN, DT , JJ , NN is most common Quad gram pos tag in all classes ,also DT , JJ, NN, IN is second most common Quad gram pos tag in all classes.

hence, can't act as important parameter for classification.

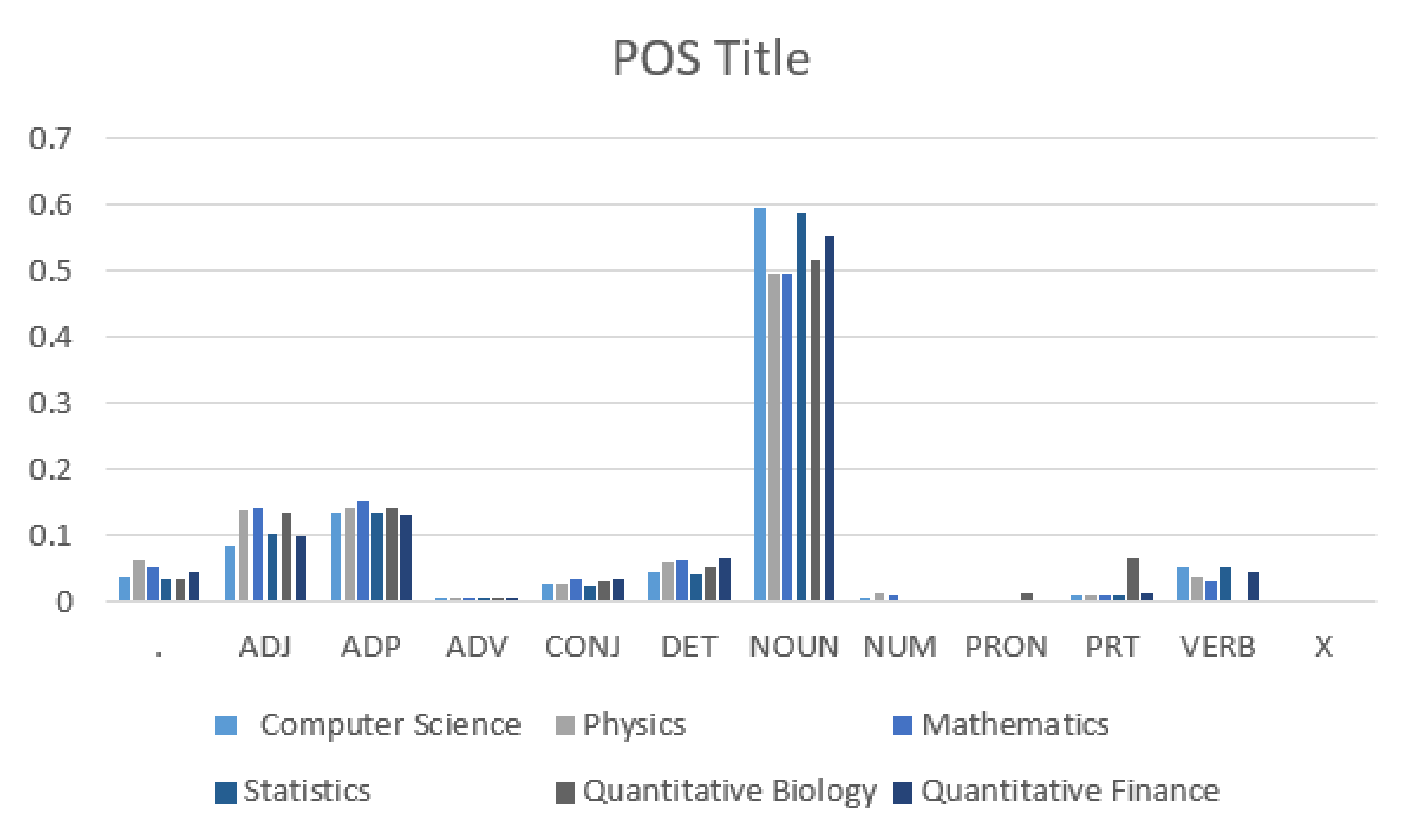
3.2 Linguistic Features and Feature Selection

POS tag important ratio calculation , to check for feature for classification

Figure 5: POS tag Title

	Computer Science		Physics		Mathematics		Statistics		Quantitative Biology		Quantitative Finance	
	POS	COUNT	POS	COUNT	POS	COUNT	POS	COUNT	POS	COUNT	POS	COUNT
	VERB	4196	ADP	10008	ADJ	7728	NOUN	28318	.	230	NOUN	1394
	ADJ	6863	DET	4229	NOUN	27060	ADP	6535	ADJ	883	ADJ	251
	NOUN	48081	NOUN	34824	CONJ	1819	CONJ	1184	NOUN	3355	ADV	13
	ADP	10877	CONJ	2020	ADP	8301	PRT	475	NUM	19	.	111
	CONJ	2119	ADJ	9604	DET	3394	VERB	2548	PRT	81	DET	171
	PRT	802	NUM	992	.	2861	ADJ	4971	VERB	420	ADP	333
	DET	3605	.	4504	VERB	1756	DET	1959	DET	341	CONJ	87
	.	3093	PRON	185	NUM	474	.	1675	ADP	912	VERB	117
	NUM	339	VERB	2732	PRT	581	PRON	116	CONJ	189	PRON	7
	PRON	192	PRT	582	ADV	389	ADV	249	ADV	38	PRT	34
	ADV	388	ADV	398	PRON	141	NUM	122	X	7	NUM	5
	X	28	X	29	X	31	X	12	PRON	16	X	2

Figure 6: POS tag Title graph



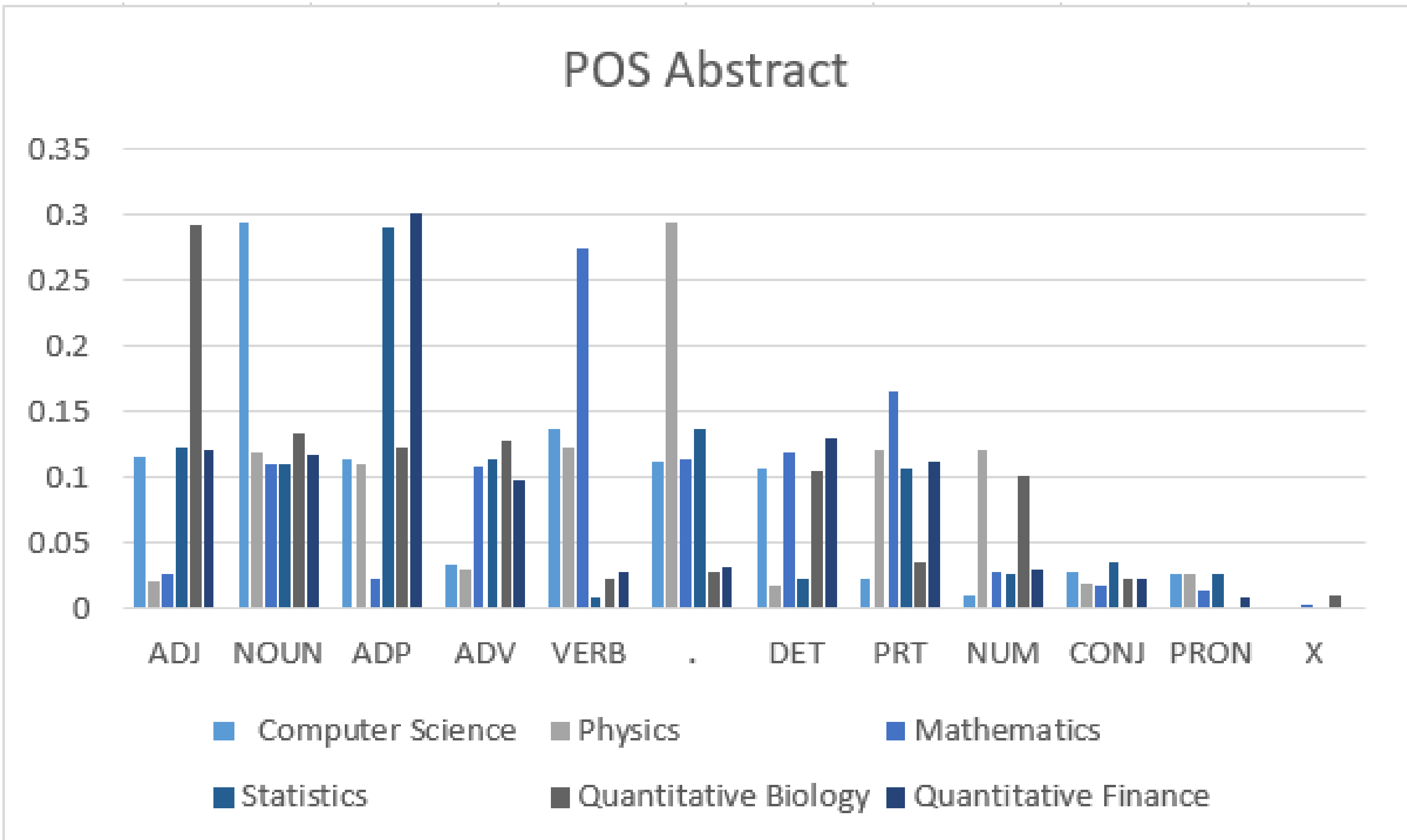
Remark

- Noun is most common pos tag.
- So, we can use top 1000 noun as a word vector and use them as a important parameter for classification.
- physics and mathematics has more cardinal tag than others.
- biology have much more number of pronouns tag than others.
- biology have much more number of particles tag than others.
- biology have much less number of verbs tag than others.

Figure 7: POS tag Abstract

Computer Science		Physics		Mathematics		Statistics		Quantitative Biology		Quantitative Finance	
POS	COUNT	POS	COUNT	POS	COUNT	POS	COUNT	POS	COUNT	POS	COUNT
ADJ	180830	PRON	23406	PRON	20375	ADJ	112779	NOUN	34256	ADP	4733
NOUN	456594	VERB	133195	VERB	88118	.	101295	VERB	15654	DET	4560
ADP	177478	DET	122916	CONJ	17583	NOUN	265968	ADP	14264	NOUN	11804
ADV	51622	ADV	33523	DET	86586	ADP	104925	ADJ	14924	.	3795
VERB	213031	ADJ	138327	NOUN	219221	NUM	7957	PRT	2582	PRON	1100
.	175045	NOUN	329950	ADP	91096	VERB	125262	CONJ	3320	ADV	1220
DET	164597	NUM	20033	ADJ	95160	PRT	21031	.	12149	VERB	5068
PRT	36143	.	135218	.	132709	DET	96949	ADV	4105	ADJ	4381
NUM	15129	ADP	134798	ADV	22057	CONJ	24288	DET	11745	CONJ	1138
CONJ	42833	PRT	21470	PRT	13175	ADV	32254	PRON	2723	PRT	908
PRON	40455	CONJ	29181	NUM	10955	PRON	24339	X	63	NUM	334
X	1258	X	769	X	1575	X	652	NUM	1144	X	10

Figure 8: POS tag Abstract graph



3.3 POS analysis

3.3.1 POS Ratios Analysis

Remark

- **Adjective by Pronoun ratio**
  physics has highest Adjective by Pronoun ratio , where as finance has least.

- **Adjective by Verb ratio**
  physics and maths has highest Adjective by Verb ratio , rest all are approximately same.

- **Adverbs to Adjective ratios**
  statistics has highest Adverbs to Adjective ratios ,physics and maths has least Adverbs to Adjective ratios.

- **Adverbs to Nouns ratios**
  statistics has highest Adverbs to Nouns ratios.

- **Adverbs to Nouns ratios**
  biology and statistics has highest Adverbs to Nouns ratios.

- **verbs to Pronouns ratios**
  biology has highest verbs to Pronouns ratios, where as maths and finance has least.

- **Cardinals to Nouns ratios**
  mathematics has very very high Cardinals to Nouns ratios where as finance is very very low Cardinals to Nouns ratios.

- **Nouns to Pronouns ratios**
  physics has highest Nouns to Pronouns ratios.

- **Verbs to Pronouns ratios**
  biology and physics has high Verbs to Pronouns ratios where as mathematics and finance has low Verbs to Pronouns ratios.

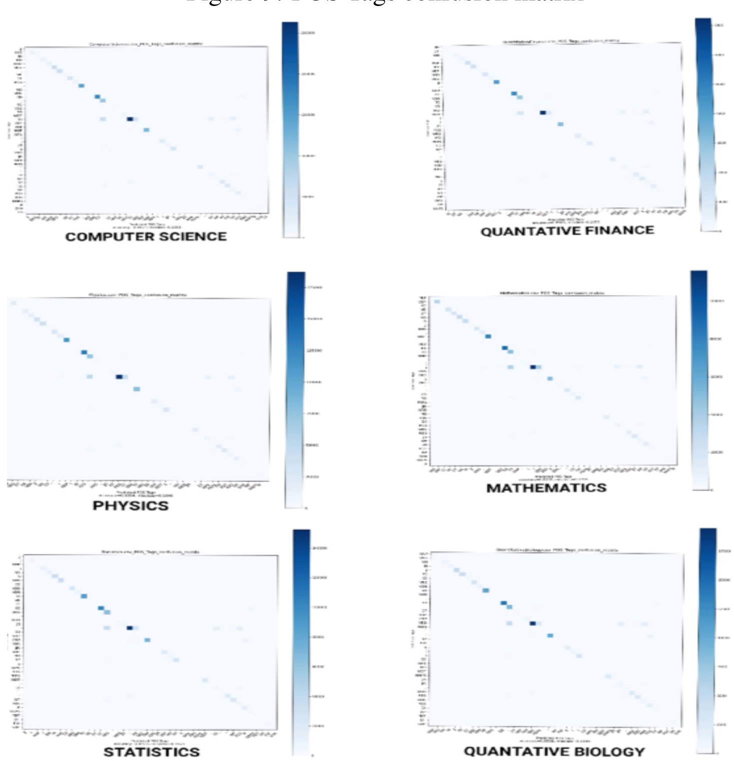
3.4 Lexical Ambiguity

Detecting ambiguity using pos confusion matrix.

Confusion matrix: In the following section, we looked at the abstracts of all the papers and used NLTK on it to get the pos tags for the same. These pos tags become our test tags. Then we ran the pos tagger for each word individually and that becomes our true tags. Using this data, we then generated the confusion matrices that you can see below.

we have calculated pos confusion matrix for each Corpus (class) and stored into data folder in png format. Python code for doing this in ***confusion matrix.py***

Figure 9: POS Tags confusion matrix



4 Conclusion:

In this project, our main goal was to identify linguistic features that can be used to classify research papers, into their corresponding categories. Our major focus has been on the sequencing and analysis of POS tags. From the data that we got above, it is safe to say that pos tags might not be a good method to try to extract information that might help us in

Type	Features
Low-level ratio	Type token ratio
High-level ratio	Adjective by Pronoun ratio Adjective by Verb ratio Adverbs to Adjective ratios Adverbs to Nouns ratios Adverbs to Nouns ratios verbs to Pronouns ratios Cardinals to Nouns ratios Nouns to Pronouns ratios Verbs to Pronouns ratios

Table 2: Derived linguistic features

multiclass classification. Although, it can be useful to perform some binary classifications like Maths vs. non-maths and so on. Nevertheless, it was a great experience for us to work on this project and we would like to thank Dr. Rajakrishnan and Siddharth sir for guiding us through the project.

5 Future work

- Implementation to actual multi class classification using important parameters we got in this whole analysis.
- Study psycholinguistic behaviour of why some specific POS is high in occurrence in a specific class.
- Study psycholinguistic behaviour of why some specific POS ratio is high in a specific class.
- Finding specific word from confusion matrix which lead lexical ambiguity.

References

[1] "Qureshi, Mohammed Rameez and Ranjan, Sidharth and Rajkumar, Rajakrishnan and Shah, Kushal" A Simple Approach to Classify Fictional and Non-Fictional Genres In *Association for Computational Linguistics*, pages "81–89". aclweb, 2019.