Natural Language Processing Project Report

Dogan Can Demirbilek

May 2021

1 Introduction

With the large volumes of scientific papers, finding relevant ones for a particular research topic can be daunting task. An approach that helps to tag or classify these scientific papers into their topic can ease the search and retrieval process and it can improve the discoverability of the documents. This study presents NLP based solutions and their comparisons to classify the given scientific papers' title and abstract into 6 topics (Computer Science, Physics, Mathematics, Statistics, Quantitative Biology and Quantitative Finance) as well as NLP based interpretations to have better understanding on the data [1] and on the methods that are used.

As in any data driven problems, descriptive analysis of the data and data preprocessing steps are implemented to understand the structure of the data and to minimize the variation in the text. After that, two approaches MultiOutputClassifier and ClassifierChain from [2] are used with logistic regression. Since logistic regression is easy to interpret, by adapting this interpretability to mentioned approaches, further understanding on data is acquired as well as deeper understanding on the approaches. As final step, proper metrics are used to evaluate the performance of these approaches.

Second approach is using deep learning model. Simple CNN architecture is used after data mapping process and similar evaluation method and metrics are used as previous approaches to have general comparison among the methods.

Lastly, semi-supervised approach Guided LDA is used with same number of topics to discover if given topics intersect with topics which are found by LDA.

2 Descriptive Data Analysis and Data Preprocessing

The raw data contains title and abstract of the paper and its labels. There are around 21k articles in the data. Some interesting properties of the data are around 5k articles have more than one label and some topics are really common whereas some of them are rare.

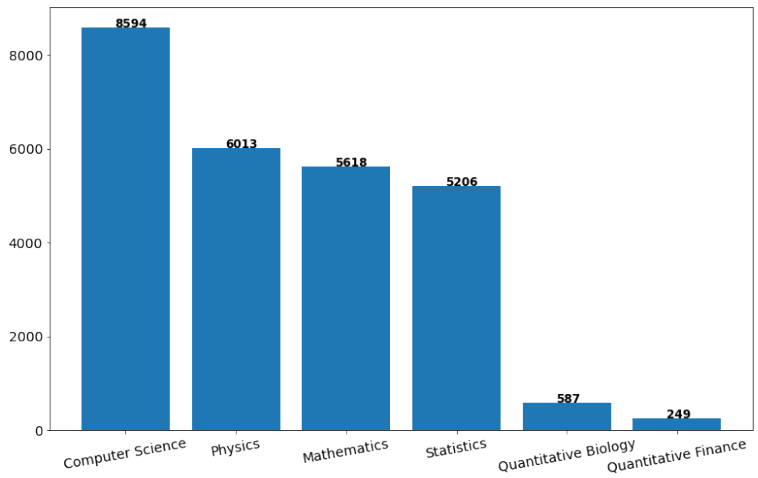

Figure 1: Count of Topics in Data

However, as it is mentioned before, some articles has more than one label, so with simple manipulation we can observe that there are 24 unique labels. Again in this case some of the topics are so rare, especially the ones with 3 labels.

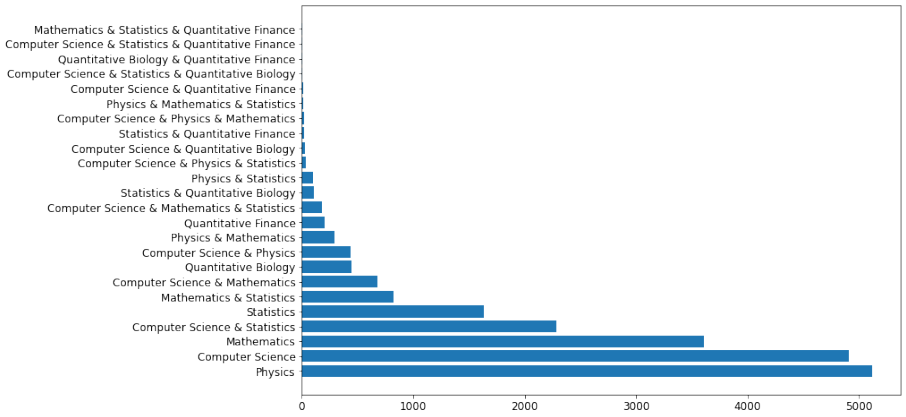

Figure 2: Count of Unique Topics in Data

Before starting to minimize the variations, firstly, title and abstract columns are combined to have single column. Each topic has its own binary column in data so values of those columns are combined in single column. It is worth mentioning that most of these articles are written in Latex so the complex preprocessing will be latex related, rest is common for any NLP task. During the random sampling from data (to get familiar with the data), it is realized that text has too many new line escape sequence, this and dashes are replaced with space.

Text is lower cased, numbers are removed. Any mathematical equations (can be composed by different tags and symbols in latex) are replaced with mathexpr by using regular expressions. Any latex related tags and punctuation are removed. Words are lemmatized and only nouns, verbs, adjectives, adverbs and proper nouns are kept. It is worth mentioning that preprocessing is updated during the study to have better results. One final step is combining some of the words into single one as an intuition to help models for better classification (e. g neural network to neuralnetwork, because biology papers include network word in different content).

Before removing the common English stop words, simple procedure is followed to find some field specific stop words. Because there could be common terminology for scientific community which is likely. To detect those words, the following procedure is applied. Wordcloud of each topic is created, it is expected to see those words (commonly used for all topics) for each topic cloud. Also most common words in the collection are found, they are big candidates to be stop words. Lastly 10 most common words are found for each topic and they are intersected in set fashion to find the common ones. By using these approaches and with the help of intuition, field specific stop words are removed as well as regular ones.

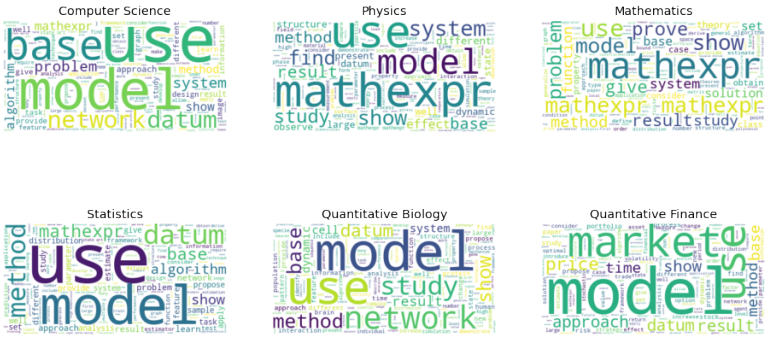

Figure 3: Word Clouds for each Topic

From Figure 3, we can see that words like use, model, method and result seem like field specific stop words.

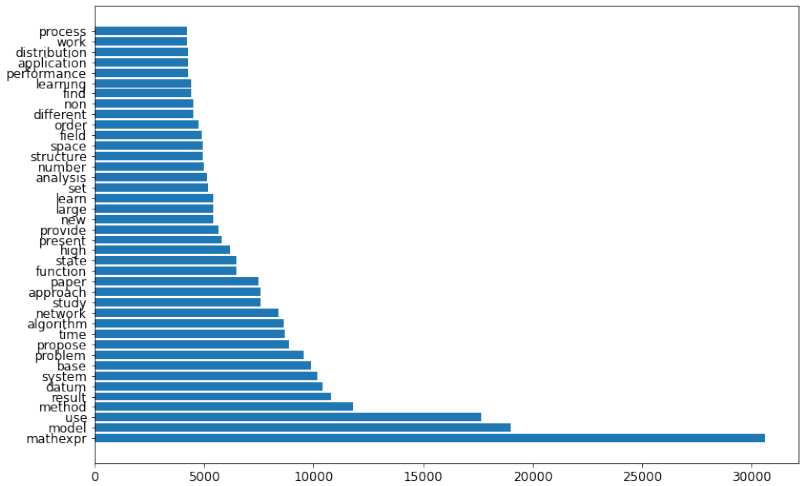

Figure 4: Most Common Words

From Figure 4, we can see the similar findings as Figure 3. Also mathexpr is the most common word after preprocessing. However this is because of some of the most common topics like Physics, Mathematics and Statistics have a lot of mathematical expression. Ten most common words are found for each topic. Result of last step is similar to findings from Figure 3 and Figure 4. Words use, model are common for all topics followed by mathexpr, method, result and base. At the end, words: use, model, method, result, study, paper, base, approach, find, problem, provide and propose are removed by following the previous procedures as well as intuition. Word mathexpr aren't removed because there is a previous belief that this word may help to classify the Mathematics related articles because they are most likely to include this term more than others.

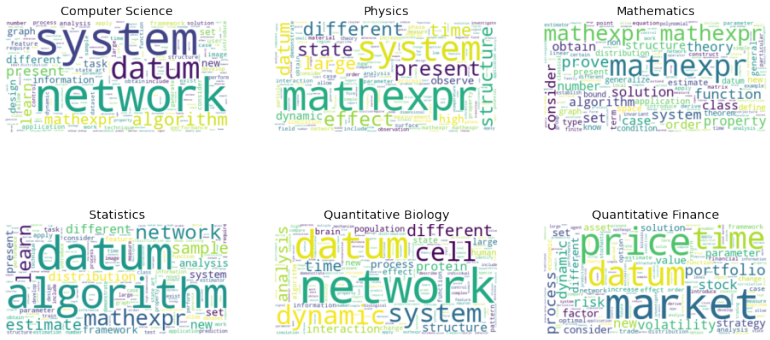

Figure 5: Word Clouds for each Topic After Stopwords Removal

3 Supervised Solution Approaches

Two main supervised approach is used to solve the problem at hand. First approach includes the MultiOuputClassifier and ClassifierChain from [2]. Although they are quite similar, there is a slight difference among them. Second approach is deep learning by using convolutional neural networks followed from [3].

3.1 Sklearn Multi Output Algorithms

For the following approaches, preprocessed text data is transformed by using TfidfVectorizer implementation from sklearn to have numerical representation of the data. Logistic regression is used with balanced for class weights parameter, because of the unbalanced data that we have.

3.1.1 MultiOuputClassifier

This strategy consists of fitting one classifier per target [2] so we ended up 6 logistic regression. Drawback of this approach is, it doesn't consider the relation among the classes.

After fitting these classifier, we can interpret the coefficients of the words to understand the contribution of the particular word to predictions. Table 1 provides the coefficient of some words for each classifier named as its target class. It can be seen that results are like expected. mathexpr has positive coefficient for Mathematics, distribution for Statistics, cell for Biology, economy for Finance and velocity for Physics.

	mathexpr	distribution	cell	economy	velocity
Computer Science	-1.05	-0.54	-1.29	-0.09	-0.71
Physics	-1.27	-0.30	-1.16	-0.52	2.52
Mathematics	2.13	0.52	-1.02	-0.36	-1.34
Statistics	-1.21	3.80	-1.29	-0.61	-1.19
Quantitative Biology	-2.90	1.16	6.83	0.84	-1.40
Quantitative Finance	-3.52	-0.09	-1.32	3.81	-0.57

Table 1: Coefficients of Some Words for Each Classifier

Evaluation of a multi-label classification is difficult because these classifications have an additional concept called partially correct. Some of the following metrics ignore this concept (exact match ratio), some of them take into consideration (accuracy). Precision, recall and F1 is calculated with macro averaging technique in estimation phase but in test data all averaging techniques (micro, macro and weighted) are reported. For estimation of these metrics, k-fold cross validation is used with different k values. Lastly, average of these values are calculated to have single estimation for each metric (Table 2).

Accuracy	Exact Match Ratio	Precision Macro	Recall Macro	F1 Macro
0.78	0.63	0.68	0.82	0.74

Table 2: Evaluation Metrics of MultiOutputClassifier with K-fold CV

K-fold cross validation is used with F1 Macro to find the best regularization parameter (C), and $C = 10$ gave the best result. Later to have better performance, best features are selected by using chi-squared stat and same metric is used with k-fold cross validation to compare the results. There is no big differences for the metric (F1 Macro) so following procedures are performed with same number of features.

For understanding the descriptive words of each topic, 10 highest coefficients are found and plotted. Figure 6 shows the results for each topic. Results are as expected, for example Computer Science has robot, language and algorithm, Quantitative Biology has brain, cell and protein etc.

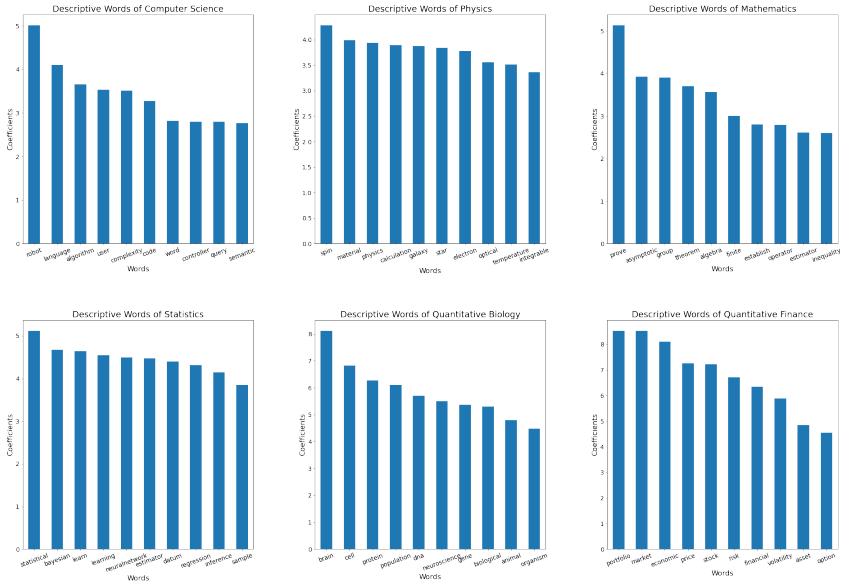

Figure 6: Descriptive Words for Each Topic

Last part includes the test of the model. For this purpose data is splitted to train, test sets to have the most possible stratified version. After training the MultiOuputClassifier, results are evaluated on the test data. Metrics (Precision Macro, Recall Macro and F1 Macro) are a bit overestimated for recall and F1 with cross validation. In general recall for all classes are higher than precision and for the classes Computer Science, Physics, Mathematics and Statistics results are better than Quantitative Biology and Quantitative Finance, most probably due to the fact that having smaller support value for Biology and Finance.

	precision	recall	F1-score	support
Computer Science	0.81	0.88	0.85	1719
Physics	0.88	0.89	0.88	1202
Mathematics	0.74	0.87	0.80	1110
Statistics	0.72	0.87	0.79	1032
Quantitative Biology	0.40	0.69	0.50	118
Quantitative Finance	0.52	0.66	0.58	50
micro avg	0.77	0.87	0.82	5231
macro avg	0.68	0.81	0.73	5231
weighted avg	0.78	0.87	0.82	5231
samples avg	0.83	0.90	0.84	5231

Table 3: Classification Report of MultiOutputClassifier on Test Data

3.1.2 ClassifierChain

Another approach to deal with multi-label classification problem is ChainClassifier. It is quite similar to MultiOutputClassifier with slight difference. This strategy is capable of exploiting correlations among the targets. In this approach, N binary classifiers are assigned an integer between 0 and N-1 where N is the number of classes. These integers define the order of models in the chain. Each classifier is then fit on the available training data plus the true labels of the classes whose models were assigned a lower number. [2] Interpretation and evaluation of the approach are done same as previous one. They show very similar results with slight differences. Therefore apart from comparison metrics, same plots won't be showed (can be found in notebook). Coefficients of same words (mathexpr, distribution, cell, economy and velocity) are checked. They show same behaviour for the topics with different values. Again k-fold cross validation is used to estimate the metrics, results can be found in Table 4. Exact match ratio and precision macro are slightly higher than MultiOutputClassifier, on the other hand, recall macro and F1 macro are smaller.

Accuracy	Exact Match Ratio	Precision Macro	Recall Macro	F1 Macro
0.78	0.66	0.71	0.67	0.67

Table 4: Evaluation Metrics of ChainClassifier with K-fold CV

Best regularization parameter is (C) 10 like previous method. Best features are selected to have better performance but like previous method there is no big difference so all the features are used for the following processes. Figure 7 shows the descriptive words for each topic, and we can observe that it is quite similar with the previous approach. Order of words are different with respect to previous one and there are some differences also for words. However all words can be related with the corresponding topic.

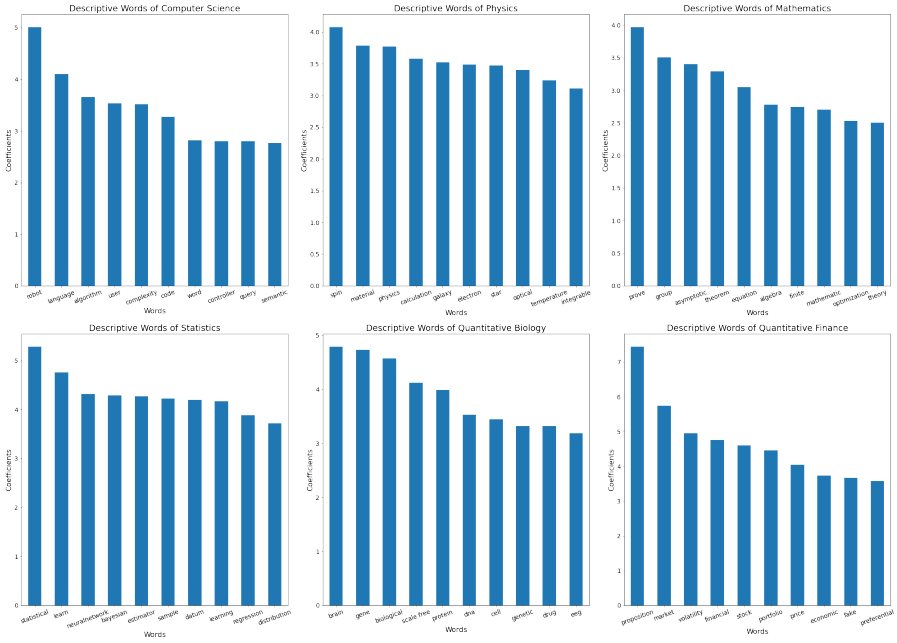

Figure 7: ChainClassifier Descriptive Words for Each Topic

Same train and test data are used to assess the performance of the approach. Table 5 presents precision, recall and F1 on test data as previous approach. Precision macro, recall macro and exact match ratio are same as cross validation results, accuracy is bit higher than cv results which is 0. 79 as well as F1 Macro which is 0. 68.

	precision	recall	F1-score	support
Computer Science	0.81	0.88	0.85	1719
Physics	0.88	0.88	0.88	1202
Mathematics	0.77	0.83	0.80	1110
Statistics	0.72	0.85	0.78	1032
Quantitative Biology	0.43	0.25	0.31	118
Quantitative Finance	0.65	0.34	0.45	50
micro avg	0.79	0.84	0.82	5231
macro avg	0.71	0.67	0.67	5231
weighted avg	0.79	0.87	0.81	5231
samples avg	0.83	0.87	0.83	5231

Table 5: Classification Report of MultiChainClassifier on Test Data

In general we can see that there is a small improvement on precision especially for the classes with low support. However there is rapid decrease in

recall for the same classes. These classes has low support and they show rapid changes to different implementations. Collecting more data for these topics could be good strategy to eliminate this vulnerability.

3.2 Deep Learning

Recently, deep learning implementations dominated the NLP related tasks. In this part, simple CNN architecture is used to deal with the problem at hand. Firstly, class weights are calculated to use in fitting phase since we have unbalanced dataset. Following formula is used to calculate these weights: $w_j = n\_samples/(n\_classes \times n\_samples_j)$ [4] where:

- w_j : is the weight for class j
- $n\_samples$: is the total number of samples or rows in the dataset
- $n\_classes$ is the total number of unique classes in the target
- $n\_samples_j$ is the total number of rows of the respective class

After this step, to have the unique integer representation of the words including padding (represented as 0) and unknown words (represented as 1), dictionary is created from train set. All words are mapped to their integer representation in both train and test sets. 95th percentile of the training sentence length is found which is 148 so shorter sentences are padded until they have this length.

Following parameters are set before starting the train phase, embedding dimension is 64, filter length is 3 and batch size is 64. Please note that these parameters are changed to have better performance on test set. However, no improvement is observed with these trials so same parameters are used from the laboratory notebook (Lecture 10 - Convolutional Neural Networks and BERT). Number of epochs are found by observing validation (%10 of the train set) categorical accuracy (Figure 8). Simply best epoch is found when it started to decrease or to stabilize. We can see from Figure 8 that after epoch 5 there is no more improvement in validation categorical accuracy so it is enough to train model with 5 epoch.

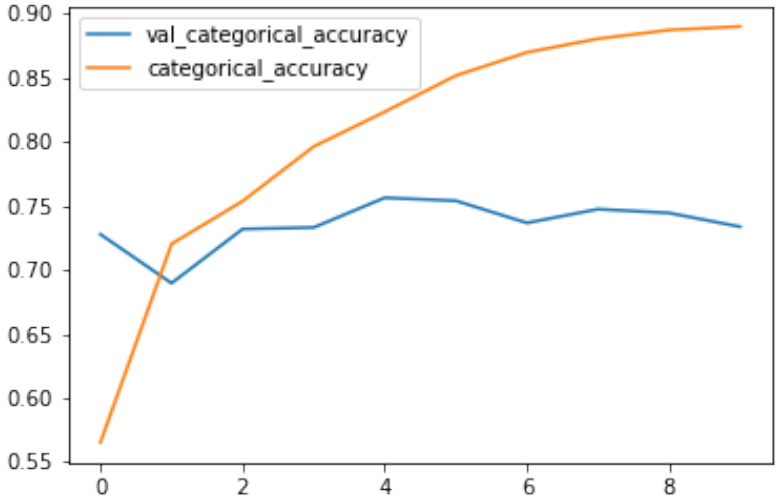

Figure 8: Train Validation Categorical Accuracy

It is important to mentioned that sigmoid activation function is used in last layer in order to deal with multi label classification problem [3]. For prediction, 0. 5 is used as a threshold, if the probability exceed this threshold, given scientific paper's topic predicted as the particular topic. Table 6 presents the performance of the model on test data.

	precision	recall	F1-score	support
Computer Science	0.84	0.77	0.80	1719
Physics	0.89	0.84	0.87	1202
Mathematics	0.83	0. 78	0.81	1110
Statistics	0.76	0.69	0.73	1032
Quantitative Biology	0.54	0.48	0.51	118
Quantitative Finance	0. 59	0. 48	0.53	50
micro avg	0.83	0.76	0.79	5231
macro avg	0.74	0. 67	0.71	5231
weighted avg	0.83	0.76	0.79	5231
samples avg	0.87	0.80	0.78	5231

Table 6: Classification Report of CNN on Test Data

It is obvious that, similar to previous approaches model perform better on the topics Computer Science, Physics, Mathematics and Statistics than Biology and Finance. Because, dataset contains less data for Biology and Finance.

Unlike previous approaches, precision and recall seems closer to each other for each topic.

3.3 Comparison

Table 7 shows the comparison among the supervised approaches. All the models performed better for the topics that have more data (Computer Science, Physics, Mathematics and Statistics) whereas they showed poorer performance on Biology and Finance. Best strategy would be to collect more data for Biology and Finance. Deep Learning and Chain Classifier perform better on precision, on the other hand MultiOutputClassifier outperform others on recall.

Supervised Approaches	Average Strategy	precision	recall	F1-score
MultiOutputClassifier	micro	0.77	0.87	0.82
	macro	0.68	0.81	0.73
	weighted	0.78	0.87	0.82
	samples	0.83	0.90	0.84
ChainClassifier	micro	0.79	0.84	0.82
	macro	0.71	0.67	0.67
	weighted	0.79	0.87	0.81
	samples	0.83	0.87	0.83
Deep Learning	micro	0.83	0.76	0.79
	macro	0.74	0.67	0.71
	weighted	0.83	0.76	0.79
	samples	0.87	0.80	0.78

Table 7: Comparison of Supervised Approaches

Performance might increase by using embedding methods which keeps the content like word2vec or by using more complex models like BERT.

4 Guided LDA

So far problem is evaluated as supervised classification problem. However, it is closely related to topic modeling. One of the most recent method to model the topics is Latent Dirichlet Allocation. It can be beneficial to guide give direction to LDA to discover the topics. For this purpose obviously same number of topics are tried to find. Topic seeds are given by intuition as well as previous approaches (give the words which have higher coefficient).

- LDA_Computer_Science : robot, complexity, algorithm, code, graph.
- LDA_Physics : spin, material, physics, electron, galaxy, star, optical, temperature.

- LDA_Mathematics: prove, asymptotic, theorem, equation, algebra, mathematics, theory.
- LDA_Statistics: statistical, learning, estimator, sample, bayesian, regression, distribution, inference, statistics.
- LDA_Biology: brain, gene, biological, protein, dna, cell, drug, population, food, human.
- LDA_Finance: market, volatility, financial, stock, portfolio, price, trading, economic, forecast, risk.

After vectorizing text by counts, guided LDA model is fitted. It provides good results for the first 4 topics (Computer Science, Physics, Mathematics and Statistics). However for the last two topics (Quantitative Biology and Quantitative Finance), it doesn't perform well. But it is interesting to point out that, for Biology we have mostly deep learning related words. It looks like, the exploding popularity of Deep Learning affected this dataset too.

- Topic LDA_Computer_Science: algorithm, graph, time, optimization, optimal, code.
- Topic LDA_Physics: field, energy, phase, state, high, spin.
- Topic LDA_Mathematics: mathexpr, mathexpr mathexpr, theory, equation, space, function.
- Topic LDA_Statistics: datum, learning, distribution, sample, estimate, parameter.
- Topic LDA_Biology: learn, deep, image, task, neuralnetwork, network.
- Topic LDA_Finance: network, datum, time, user, analysis, information.

Lastly, it is interesting to check the coverage of LDA topics of the true labels (Figure 9).

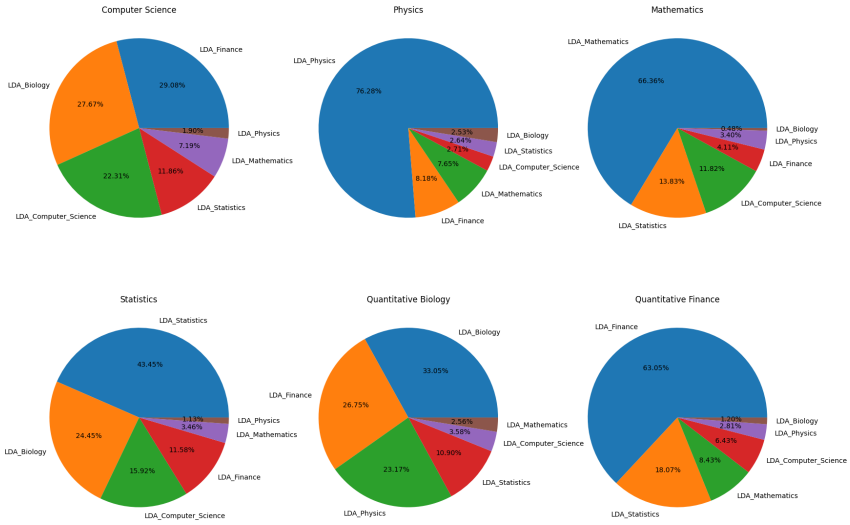

Figure 9: Coverage of LDA Topics

We can observe that Computer Science is mostly covered by LDA_Finance and LDA_Biology, looks like LDA find subtopics of Computer Science and explains them by using Biology and Finance which is parallel to deep learning interpretation that is made. Same situation seems valid also for Statistics. It is also interesting to interpret the results in terms of proximity of the topics. For example, recently (due to the machine learning researches) Computer Science and Statistics articles are overlapped. Also, it seems like Mathematics and Statistics are close to each other which make sense.

5 Conclusion

In this study, NLP based solution and their interpretations are presented for multi label classification problem. First, data explanatory analysis is performed to have understanding on data and preprocessing are implemented to minimize the variance in the text. After that, three supervised approaches are presented to solve the problem at hand. Also interpretation of the first two approaches are presented to have deeper understanding on the methods and the data. Comparison of the supervised approaches are made and strong and weak points of these approaches on this problem are presented. Lastly, guided LDA is used to have better understanding on the structure of the data as well as relation of the topics.

References

[1] "Multi-Label Classification Dataset". In: (2021). URL: https://www.kaggle.com/shivanandmn/multilabel-classification-dataset.

[2] "Multiclass and multioutput algorithms". In: (2021). URL: https://scikit-learn.org/stable/modules/multiclass.html#multilabel-classification.

[3] "Performing Multi-label Text Classification with Keras". In: (2021). URL: https://blog.mimacom.com/text-classification/.

[4] "How to Improve Class Imbalance using Class Weights in Machine Learning". In: (2021). URL: https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/.