Venue Detection For Research Papers

Syed Ayaz
ayaz01@ads.uni-passau.de
University of Passau, Germany

ABSTRACT

There are lots of scientific papers published in different fields of research everyday and authors have choices to publish them in different conferences and venues. Authors and writers get confused while deciding which venue is most suitable for publishing their new paper. Our project deals this issue in a precise way using different data science techniques. In this report, we have presented the road map to solve citation based venue detection. A research paper is taken with properties like abstract, title, keywords and citation. Then by processing this data, a suitable venue is suggested where author can publish his/her newly written paper.

The project is divided into four phases. The first phase is about designing the whole scenario which explains the architecture of project and work-flow of all tasks. In second phase, different pre-processing techniques are discussed and how these techniques are applied on the data-set. A Tf/IFD [12] vectorizer is created after applying the preprocessing techniques. Third phase covered the analysis part in which different classification and clustering techniques have been explored. In last phase confusion matrix [16]is used to compare the results in detailed way. It also evaluates the results of different classifiers by changing the parameters.

INTRODUCTION

The main area of our work is venue detection for new or existing scientific research papers. With the growing number of research papers, it may become problematic to find out a best venue to present the paper in. This work is about predicting a venue that is most relevant for the research topic under consideration, so that this research can be presented and shared in the most appropriate audience. Such a utility is extremely useful for both the writer as well as the literary community. This is not a time based problem so the time series can not be applied. The nature of this problem is clearly classification since the available data can be best used to classify based on existing data. Supervised classification is used to predict a certain class for the paper content. We have chosen supervised classification as all the venues are already available in the data-set so there is no need to collect all the possible venues. Various machine learning techniques have been used with special focus on natural language processing since this is a text based problem.

0.1 Problem statement

After writing a research paper, it can be a tedious job to find out which venue would be best suited for a research paper which is going to be published. It can be a problem specially if you do not have enough knowledge about all the possible venues. The main purpose of our work is to be able to predict the best venue for a research paper based on its research content. This will help the new content writers as well as the experienced ones to present their work on a specific and more relevant venue.

0.2 Approach

Our approach is rather simple in regards to the problem at hand. Since we are using classification, we have all the classes at hand in the data-set. Thus the process of classification can be done quite easily with the help of a TF/IDF matrix. From all the available features, Title and Abstract are the most important ones. By creating the terms from Title + Abstract + Keywords and documents from Title + Abstract, the created matrix is used to classify the unknown papers.

For a start, citation data-set is taken from the corpus and filtering is applied on it, by which title, keyword and and abstract are selected. After applying preprocessing steps including removal of all NaN values, tokenization and stemming of Title, Abstract and Keywords columns, a data-set is produced with the cleaned and processed data. In next step, terms for TF/IDF[12] are composed of most repetitive distinct words of Title and Abstract. This TF/IDF is then used by classifiers in next step as X-input. For Y-input, venues of cited paper are given. The results of classifiers are evaluated by using confusion matrix[16] and are shown with the help of graphs and charts. Figure-1 shows an over all design of the entire approach of our detection model.

1 DATA ACQUISITION & PRE-PROCESSING

1.1 Acquisition

To solve this particular problem, the citation data is taken from the readily available online source using the link [1]. The data set is designed for research purposes only. The citation data is an amalgamation of research papers which are extracted from LISP, DBLP, ACM, MAG (Microsoft Academic Graph), and other sources. Currently, we are using V11 [2] which contains 4,107,340 papers and 36,624,464 citation relationships as of . The reason for choosing this particular data-set among all the other available data-sets (V1 to V10) is the fact that this data-set offers a large variety of fields and more data to filter from, in order to make the classification more accurate.

Each paper incorporates a couple of different fields which can be seen in the Table-1.

As documents are comprised of unstructured text which holds obscure information so, data pre-processing is applied which consist of the essential steps for converting unstructured data to structured data. The steps performed in the pre-processing phase are as follows:

[1] https://aminer.org/citation
[2] https://lfs.aminer.cn/misc/dblp.v11.zip

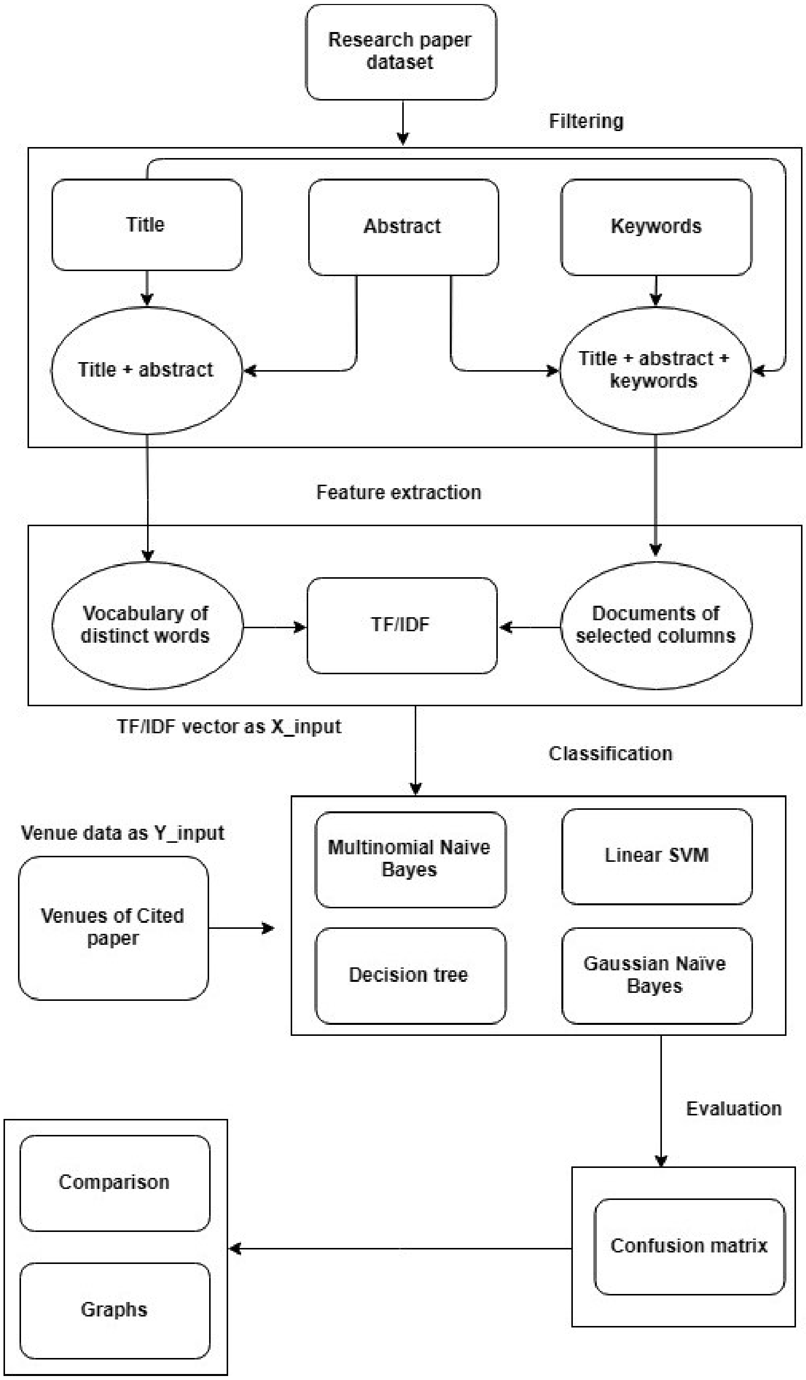

Figure 1: Architectural Design For Venue Detection
The figure displays an over all work-flow of the venue detection from beginning to the end.

- Data Pruning (filtering and grouping).
- Tokenization and Stop words.
- Stemming and Lemmatization.
- Bag-of-Words model.
- Term frequency-inverse document frequency(TF-IDF) [12]

The citation data-set is passed through the various steps of data pre-processing to get not only the structured data but also to get some useful insights and patterns in the data. This step is vital because of its purpose affects on the whole results. So, firstly, the data is pruned to filter out the fields that we actually need for the

Table 1: Data fields before filtering it into precise data

Field Name	Field Type	Description	Example
id	string	paper ID	53e9ab9eb7602d970354a97e
title	string	paper title	Data mining: concepts and techniques
authors.name	string	author name	Jiawei Han
author.org	string	author affiliation	Department of Computer Science, UOI at Urbana-Champaign
author.id	string	author ID	53f42f36dabfaedce54dcd0c
venue.id	string	paper venue ID	53e17f5b20f7dfbc07e8ac6e
venue.raw	string	paper venue name	Inteligencia Artificial
year	int	published year	2000
keywords	list of strings	keywords	["data mining", "structured data", "world wide web"]
fos.name	string	paper fields of study	Web mining
fos.w	float	fields of study weight	0.659690857
references	list of strings	paper references	["4909282", "16018031", "16159250", "19838944", ...]
n_citation	int	citation number	40829
page_start	string	page start	11
page_end	string	page end	18
doc_type	string	paper type: journal, book title...	book
lang	string	detected language	en
publisher	string	publisher	Elsevier
volume	string	volume	10
issue	string	issue	29
issn	string	issn	0020-7136
isbn	string	isbn	1-55860-489-8
doi	string	doi	10.4114/ia.v10i29.873
pdf	string	pdf URL	//static.aminer.org/upload/pdf/1254/ 370/239/53e...4a97e.pdf
url	list	external links	["http://dx.doi.org/10.4114/ia.v10i29.873"]
abstract	string	abstract	Our ability to generate...
indexed_abstract	dict	indexed abstract	"IndexLength": 164, "Inverte dIndex": "Our": [0]

task at hand.

1.2 Motivation For Fields Selection

The field selection criteria is simply the relevancy to the venue itself. Following will be explained the reason for each field selected. "Id" is needed to access a specific record in order to perform some specific processing on that record. "Tile" is necessary to get the context of the document. "Venue.raw" is needed for the "Y" parameter of the classifier in the classification phase. "Year" is needed to visualize the data in order to find out the yearly distribution of papers published. This data distribution is helpful in finding out the trends of taking the right kind of data. "Keywords" are needed to increase the information in the document. "Fos.Name" FOS stands for "Field Of Study" and this information is used in the analysis process. "n-Citation" is the number of time the paper is sighted in other papers. We have set a threshold of citation number and only the records with more than 5 value are selected. "Lang" is used to check if the paper is written in English language because we are only working with papers written in English. "Abstract" is the most important field in our situation since most of the content for TF/IDF[12] comes from this field.

1.3 Motivation For Fields Rejection

The fields that are not considered for further processing include Author.Name and Author.Org. Author data generally is not relevant to venue detection because it does not matter who the author is, the paper will be published regardless of the borders and ethnicity. Hence it contains no direct information about the venue itself. Venue.Id is not taken because the information about venues can be taken from Venue.Raw. Fos.W is the weight of field of study among all the other fields of studies and this has no apparent use in venue detection since Venue.Name serves the purpose of taking some extra keywords. Page-Start and Page-End are equally useless. Doc-Type tells about the document being of journal or book type which again does not help in venue detection. Publisher was only helpful in finding out the publisher to paper distribution which is not relevant in our scenario. Volume, Issue, ISSN, ISBN, DOI are the integer values and do not help in recognizing a pattern. PDf urls and external urls also give away no such information.

1.4 Pruning and Filtering Of Selected Fields

All of the research papers have title and abstract which contain a lot of unnecessary words and symbols that need to be eliminated for making our model fast and accurate. So to understand the content of the text at the word level, tokenization is performed on the content (title + abstract) of the data-set. It converts the sentences and phrases of the content to tokens in order to do lexical analysis. This tokanization is performed using the NLTK[14] package called nltk.tokanize.

Stop words such as "I", "me", "my", "myself" etc. are removed from the content in order to retain the words that are most important and meaningful in the sentence. The list of stop words that is removed, is taken from the NLTK's implicit stop words list which is available online[3] .

In the next step, the content is now lemmatized in order to convert the words to their base form so that it becomes easier to find word count in the document (research paper) when making TF-IDF. Once again Data is lematized using NLTK's WordNetLemmatizer[17] package which is taken from nltk.stem.

The pre-processing steps can be seen graphically in Figure.2 in which data-set is divided into sub-parts and we have applied data pruning to select only those columns which are important for our processing like abstract, title and keywords.

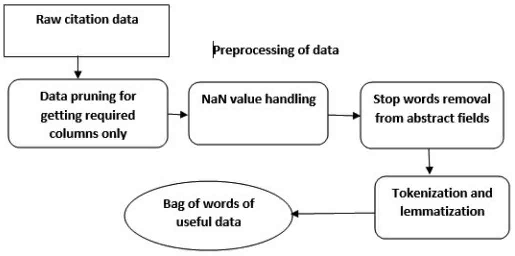
Raw citation data
Preprocessing of data
Data pruning for getting required columns only
NaN value handling
Stop words removal from abstract fields
Tokenization and lemmatization
Bag of words of useful data

Figure 2: Pre-processing of data-set

This image shows all the steps involved in pre-processing phase for venue detection of research papers.

1.5 Data Size and Memory Constraints

Due to the large amount of data, it was unfeasible to parse all the data on our machine with limited resources. The raw file was a total of 11 GB size and could not possibly be parsed on our machine. Out of 4,107,340 papers records, we had to come up with an approach to fetch as much data as was feasible. The file is available in JSON format which is loaded using the $read_json$() method given by "nltk" [14]. Initially, we tried to load the entire 11GB data through the aforementioned method which ended in memory overload error. The first approach was to just take 115000 entries from 4,107,340 papers and just work and see if it works. In the rest of report, we will refer to this data as 3Million data-set.

1.6 Feature Preparation and Merging of Multiple Features

We are using Pandas[15] DataFrame which is a very useful tool, however, it needs the data to be in good shape to work well. Most of the data in each column was initially in dictionary format. We processed each column according to what we need from the column and removed the rest in order to increase the memory efficiency. From "title", we took it as is because it was simply a string. "fos" was a dictionary with "name" as key and "w" as the weight. We only took name, since that was the data of our concern. "venue" is a list of dictionaries with each dictionary containing "id" as key and "raw" as its value. We extracted the venue names from this. "indexed_abstract" is an interesting field because it contained some confusing data at the first sight. On a closer look, it turned out to be very useful field since it contained tokenized form of abstract words along with the indices of each word. This was also a list of dictionaries with each dictionary having index as key and word as its value. "doc_type", "publisher" and "year" is also a string type, so these are taken in their original form.

1.7 Initial Data Insights

After setting the proper data types and initial data pruning, we needed to visualize the selected data in order to get a more clear understanding of the data. Following are some data insights which gave us some initial understanding of the data. From an overall point of view with regards to the papers published yearly we calculated some numerical statistics of the 3 Million data-set. We found that the earliest papers were presented in 1937, 10% of the papers were presented between 1937 to 1993, 50% of the papers were presented until 2007, 90% of the papers were presented until 2015 and over all papers were presented until 2019. This information can be seen in the table-2 below.

Statistics	Year
Mean	2005
Min	1937
10%	1993
50%	2007
90%	2015
Max	2019

Table 2: Statistics about description of years in data-set

1.8 NAN Value Handling

Next, we had to handle the NAN values among the data columns. Initially, we simply got rid of all the rows with NAN entries using Pandas[15] dropna() method. The reason for that being we had all the data we needed for the pre-processing, so we could make a little quantity over the quality compromise of data. After the removal of NAN entries, the numerical statistics on the new data shows the first paper presented in 1946 and the last one in 2019. About 40% of the papers were presented between 2011 and 2015. Only 10% of the papers were presented between 2015 to 2019. These statistics can

[3] https://gist.github.com/sebleier/554280

be seen in the table-3 below.

Statistics	Year
Mean	2008
Min	1946
10%	1999
50%	2011
90%	2015
Max	2019

Table 3: Statistics about description of years after removal of NaN values

1.9 Visualization Before Processing

A better visualization can be done by seeing the data in plots. First the data is plotted over the years which can be seen in the Figure. It is quite clear that most of the interested data comes from the year range of 2000 to 2015. This can also be verified from the numerical data distribution as explained previously. The plot can be seen in Figure-3 below.

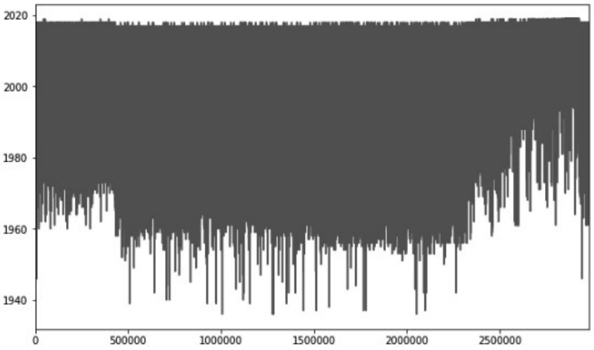

Figure 3: Distribution of data over the years before pre-processing

Next, histogram is calculated and gives the same credentials. It can also be seen from the histogram that most paper data saturation is seen between the year 2000 and 2015. The histogram can be seen in Figure-4.

Field of study (fos) is another field that gives quite useful insight. The figure shows the papers distribution written in a particular field of study. In our particular data-set, Philosophy and Performance art contains the most papers. In order to find out the papers distributed over all of the venues, a horizontal bar chart is displayed in the figure. It can be seen that about 1000 papers were published in "Lecture Notes in Computer Science".

1.10 Visualization After Processing

Next, the data is filtered to keep only the columns that are needed and the rest of the columns are discarded from 3Million data-set.

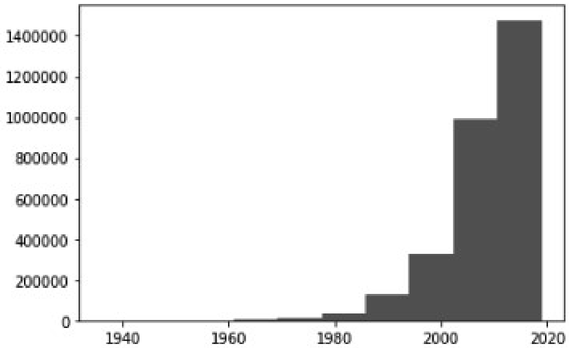

Figure 4: Distribution of data over the years before pre-processing in Histogram

papers published in a particular field of study.png

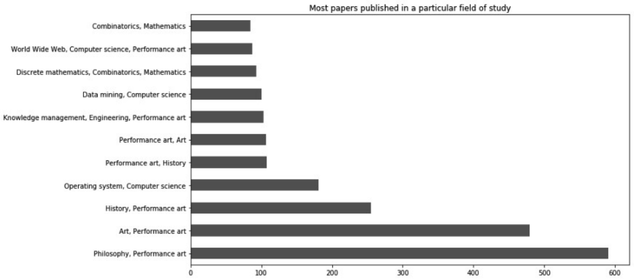

Figure 5: Number of papers published in particular domain before applying pre-processing

Of Papers Published In a particular Venue.png

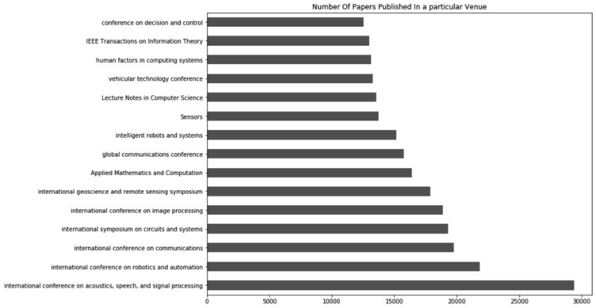

Figure 6: Number of papers published in a particular venue.

After the pre-processing steps, we ended up with 0.6Million data-set. For our ease, this filtered data is saved to a CSV file which will be read from now on to access the processed data. After the first data filtering, the data visualization also performed and can be seen in the figures below.

The data is again plotted over the years which can be seen in the Figure. It can be seen that most of the interested data still comes from 2000 to 2015. Hence the data was not changed a lot.

Histogram is also calculated again and almost gives the same of data over the year.png

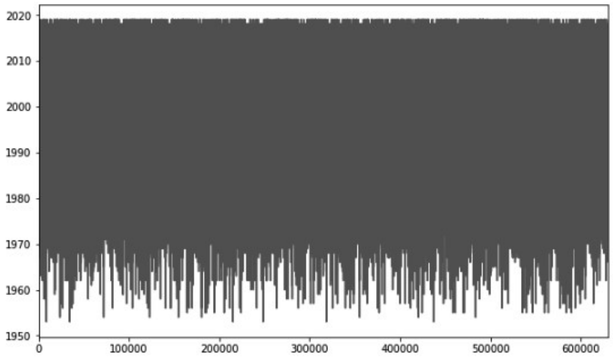

Figure 7: Papers published in a particular venue during particular years

This image shows all the steps involved in pre-processing phase for venue detection of research papers.

credentials. It can also be seen from the histogram that most data saturation is seen between 2000 to 2015.

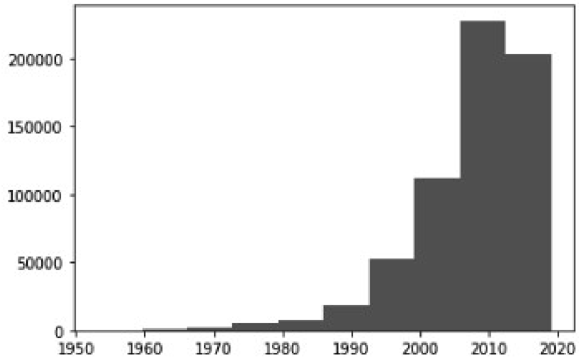

Figure 8: Distribution of data over the years before preprocessing

Figure-9 shows the papers distribution written in a particular field of study on the 0.6Million data.

Another useful information is the fields of studies that most papers are written for. Table-4 shows most frequent fields of studies in which the papers are written for and can be seen below.

The following Table-5 shows the most promising venues that are often used for paper publishing. This information can help in eliminating all the unpopular venues which may cause an overhead in classification.

papers published in a particular field of study.png

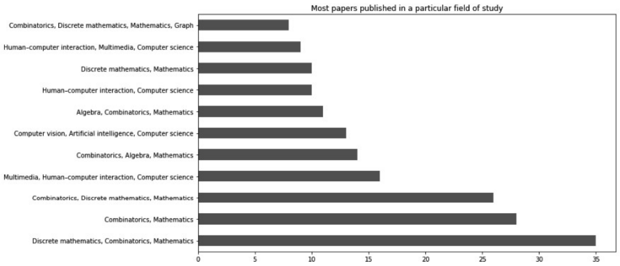

Figure 9: Number of papers published in a particular field of studies

This image shows all the steps involved in pre-processing phase for venue detection of research papers.

Field of studies	No of papers
Philosophy, Performance art	46
Art, Performance art	34
History, Performance art	22
Operating system, Computer science	20
iPerformance art, Art	14
Performance art, History	12
Computer science, Knowledge management, Performance art	6
Operating system, Art	5
Knowledge management, Computer science	4
Machine learning, Artificial intelligence, Computer science	4

Table 4: Most paper published in a particular field of study

Venues	No of papers
International Conference on Acoustics	15125
International Conference on Robotics	13048
Lecture Notes In Computer Science	12759
International Symposium on Circuits	11524
Discrete Mathematics	7441
International Conference on Pattern Recognition	5678
IEEE Transactions on Communication	5727

Table 5: Features representing most promising venues in a particular year

1.11 Special Character Removal

Next, we removed the special characters from "Title", "Abstract" and "fos" in order to improve the learning as these characters can not possibly have any valuable information with regards to venue detection. One of the draw backs of removing these characters was that the data ended up having some abandoned characters and numbers that were previously used with those special characters. These abandoned characters and numbers were removed by removing anything that has a length of lass that three characters.

1.12 Term Frequency/Inverse Document Frequency TF-IDF

It is an approach that provides dense vector representation of words that try to capture the meaning of that word. The TF-IDF[12] is the product of two weights, the term frequency and the inverse document frequency. Term frequency is a weight representing how often a word occurs in a document. If we have several occurrences of the same word in one document we can expect the TF-IDF to increase. Inverse document frequency is another weight representing how common a word is across documents [9]. If a word is used in many documents then the TF-IDF will decrease. General representation of TF-IDF is given below:

$$tfidf\ (term, document) = tf\ (term, document)\ .idf\ (term)$$

The first part of it is TF, which means “Term Frequency”. By saying it we simply mean the number of times the word occurs in the document divided by the total number of words in the document:

$$tf\ (term, document) = \frac{n_i}{\sum_{k=1}^{V} nk}$$

Where n_i is the number of times a most frequent occurring word occurs in the document, nk are total numbers of documents. The second part is IDF, which stands for “Inverse Document Frequency”, which is calculated as follows:

$$idf\ (term) = log\frac{N}{n_t}$$

Where N is total number of document n_t represents the number of documents with unique words t.

1.13 Preparation of TF/IDF

In Text mining, the most popular model to represent the document (in this case the Abstract + Title + Keywords) is by using the Bag-of-Words model which uses the documents as an aggregate of words. In bag-of-words, the words contained in the documents are features that act as documents and comprises of feature vectors which express each document.
At this point, we are all set to create our TF/IDF which is our final product for the classifier. A set of vocabulary is created from "Title" and "abstract". This set is created after applying the lemmatization on "title" and "abstract" separately. The next entity that is needed for TF/IDf[12] is the documents since it finds the frequencies of words in the documents given. A set of documents is also created by "Title", "Abstract" and "fos" and all of these are combined after performing lemmatization on them.
TF-IDF incorporates the keyword frequency and the inverse document frequency. This TF-IDF[12] serves as "x" input for the classifier. For "y" lables, we use the Venue.Raw field on which we want to classify our model. Hence, by producing the TF/IDF, we have produced the vectors that are needed for the classification process. Each vector is then used as an "x" input to the classifier. In order to generate the TF/IDF, we are using TfidfVectorizer by SciKit-Learn. Vectorizer.Fit() is used to fit the key words taken from the vocabulary generated above. Vectorizer.Transform() is used to feed the documents.

2 EXPERIMENTS (ANALYSIS + EVALUATION)

In this section, we have worked on classification as well as evaluation of data for finding required results because these two phases are overlapping with each others. The analysis part is divided into two parts which are given below:

- Selection of X and Y parameters
- Hyper parameter evolution
- Manipulating the test set size

The selection of X parameter is done by using simple approach of TF-IDF matrix of our documents (i.e are the research papers) in the previous phase. For Y parameter, we have used venues. We have applied different classifiers which are given below:

- Multinomial Naive Bayes[18]
- Linear SVM[19]
- Logistic Regression
- Random Forest[?]

The motivation for using these classifiers is to compare our approach and particularly the results to the baseline approach [1]. The figure below is showing the detail working of classifiers used in our project. Every time classifier predicts some results, they are compared and evaluated with previous results on the basis of testing set size, hyper-parameter and number of features(i.e the most frequently occurring words in the all the research papers).

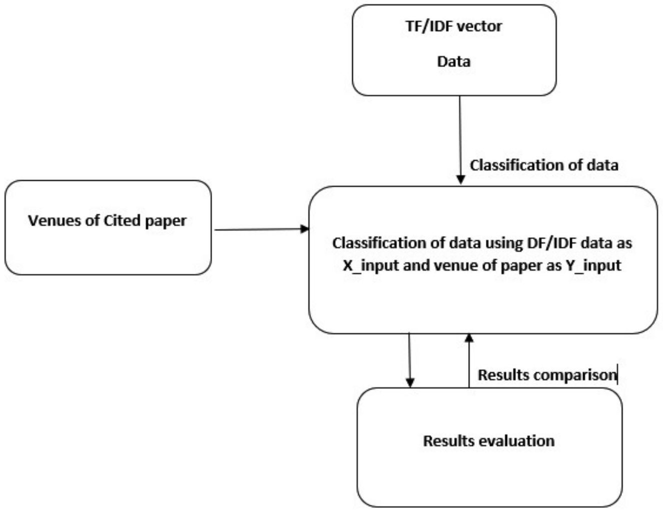

Figure 10: Visual representation of the steps of data analysis.

For the sake of getting evaluation results, we have used confusion matrix. This matrix gives us details about how the classifiers behaved and what kind of output we received after applying certain actions on our data-set. We have applied following parameters for evaluation of classification results:

- Accuracy
- Precision
- Recall
- F1-score

We have performed k-fold cross validation with the value of k being 10 as it is the most optimal value for which our classifiers gives the best results. Varying this value also helps in detecting over-fitting and choosing the right values for the hyper-parameters our classifiers.

The evaluation part is behaving like a iterative process with analysis part. To compute the accuracy of our classifiers we use confusion matrix[16] (i.e a library function of sklearn) which takes the prediction of classifiers as input and outputs the results of miss-classification and miss-rates etc. The output is also used to compute accuracy, precision, recall and f1-score. By observing and analyzing these values different variation in terms of test set size, hyper-parameter and number of features can be made to improve the overall performance of the classifiers.

2.1 Analysis part 1

We have used supervised machine learning techniques to classify our data. Supervised machine learning is a task which develops a model from labeled training data. This training data is used to estimate a mapping function f in such a way that when a new input data x is given to function, it will predict the output variable or a class y for that data input.

$$y = f(x)$$

According to the baseline paper[1], 44 thousands research papers were used as documents for the classification of 22 conference venues and for vocabulary they used which were randomly selected from different field of machine learning, computer vision, artificial intelligence, natural language processing and data mining. Their precision, recall and F1-score are 79.3%, 77.6% 78.1% followed respectively.

In the analysis part 1, we used the data-set of 65 thousands of research papers as documents. For venues we used the elbow method to find the optimal number of venues in the selected data-set. The optimal value for the date-set selected is 25. For vocabulary(i.e features values), we used the 2500 most frequently occurring words in the data-set using FreqDist[13]. We provided TF/IDf [12] vector data and venues to multinomial naive Bayes[18] and linear SVC and random forest with the default hyper-parameters to perform classification. Logistic regression had the best score among all the classifiers with 48.3% accuracy and 48.5% precision where as the second best is the linear SVC which has accuracy of 47.6% and precision of 45.5%. The reason for these classifiers outperforming all the other classifiers is because these classifiers are optimum when it comes to solving multi-class classification problems. The hyper-parameter value such as solver is set to "lbfgs" in-case of logistic regression to handle large data-sets and multinomial losses. The penalty is set to "l2". The testing set size was also varied from 10% to 30% with the optimum value that worked best for the classifiers was 20% was taken into consideration. Initially vocabulary was set to 800 feature values but the precision and accuracy of all the classifiers were low. So, by increasing the vocabulary the results were significantly improved with 50% increase in the performance of logistic regression and 45% increase in the performance of linear SVC the idea of changing the feature types.

2.2 Evaluation part 1

We evaluated that our model on the data-set of 65 thousands of research papers gave significantly low results as compared to the baseline approach. The results are given in the table below. The main reason of bad results was poor selection of features values as we had initially taken very low feature values(i.e vocabulary)for the TF/IDF [12]. Classifiers got few words(i.e vocabulary) to train on which enables them to poorly classify the papers. But as the vocabulary is increased the results got significantly better.

Classifier	Accuracy	Precision	Recall	F1-score
Linear SVC	0.476	0.455	0.453	0.445
Multinomial NB	0.259	0.290	0.1346	0.1300
Random forest	0.157	0.049	0.043	0.0169
Logistic regression	0.483	0.485	0.431	0.438

Table 6: Results of different classifiers

2.3 Analysis part 2

We used the data-set of 200 thousands of research papers as documents. For venues we used the elbow method to find the optimal number of venues in the selected data-set. The optimal value for the date-set selected is 22. For vocabulary(i.e features values), we used the 5000 most frequently occurring words in the data-set using FreqDist[13]. We provided TF/IDf [12] vector data and venues to multinomial naive Bayes and linear SVC and random forest with the default hyper-parameters to perform classification. Logistic regression had the best score among all the classifiers with 54.2% accuracy and 52.8% precision where as the second best is the linear SVC which has accuracy of 53.9% and precision of 50.7%. The hyper-parameter values for this part were used from previous optimum stage(i.e analysis part 1)

2.4 Evaluation part 2

This time we evaluated our model on the data-set of 200 thousands of research papers and Logistic regression accuracy results were improved by almost 11% whereas the linear SVC accuracy was improved by 12.3% than the previous stages. The results are displayed in the table below. The main reason of getting a little bit more accurate results was the use of more vocabulary and data-set for the classifier to train.

Classifier	Accuracy	Precision	Recall	F1-score
Linear SVC	0.539	0.507	0.496	0.495
Multinomial NB	0.431	0.504	0.305	0.310
Random forest	0.192	0.046	0.045	0.22
Logistic regression	0.542	0.528	0.489	0.497

Table 7: Results of different classifiers

2.5 Analysis part 3

We used the data-set of 4100 thousands of research papers as documents. For venues we used the elbow method to find the optimal number of venues in the selected data-set. The optimal value for the date-set selected is 60. For vocabulary(i.e features values), we used all the most frequently occurring words in the data-set using FreqDist[13]. We provided TF/IDf [12] vector data and venues to multinomial naive Bayes and linear SVC and random forest[?]. with the default hyper-parameters to perform classification. Logistic regression had the best score among all the classifiers with 85.3% accuracy and 85.3% precision where as the second best is the linear SVC which has accuracy of 82.3% and precision of 81.4%. The hyper-parameter values for this part were used from previous optimum stage(i.e analysis part 1)

2.6 Evaluation part 3

The model evaluated on the data-set of 4100 thousands of research papers and Logistic regression accuracy results were improved by almost 57.7% whereas the linear SVC accuracy was improved by 54.4% than the previous stages. The results are displayed in the table below.

Classifier	Accuracy	Precision	Recall	F1-score
Linear SVC	0.823	0.814	0.78.9	0.79.3
Multinomial NB	0.71	0.70	0.681	0.687
Random forest	0.26	0.235	0.229	0.201
Logistic regression	0.851	0.853	0.826	0.819

Table 8: Results of different classifiers

2.7 Conclusion

Deep walk is a potential method that can significantly improve the performance. However, due to some major inconsistencies in the data it is quite hard to get some meaningful improvements. This, however can be changed if the IDs of all or most the venues are available since these IDs are an important factor in creating the Deep Walk.

2.8 Future Work

Will implement Deep Walk technique after processing the data to either get the rightly probable IDs of Venues or somehow generate some meaningful IDs for them.

REFERENCES

[1] [Cornelia et .al]Cornelia Caragea, Corina Florescu: Venue Classification of Research Papers in Scholarly Digital Libraries. TPDL 2018: 129-136

[2] [Bulgarov et .al]Caragea, C., Bulgarov, F., Godea, A., Das Gollapalli, S.: Citation-enhanced key phrase extraction from research papers: a supervised approach. In: Proceedings of EMNLP (2014)

[3] [Caragea et .al]Caragea, C., Bulgarov, F., Mihalcea, R.: Co-training for topic classiïňĄcation of scholarly data. In: Proceedings of EMNLP, pp. 2357âĂŞ2366 (2015)

[4] [Giles et .al]C. Lee Giles, Kurt D. Bollacker, Steve Lawrence:CiteSeer: An Automatic Citation Indexing System. ACM DL 1998: 89-98

[5] [Sujatha et .al]Sujatha Das Gollapalli, Cornelia Caragea: Extracting Keyphrases from Research Papers Using Citation Networks. AAAI 2014: 1629-1635

[6] [Wei et .al]Wei Fan, Huan Liu, Suge Wang, Yuxiang Zhang, Yaocheng Chang: Extracting Keyphrases from Research Papers Using Word Embeddings. PAKDD (3) 2019: 54-67

[7] [Isaac et .al]Isaac G. Councill, C. Lee Giles, Min-Yen Kan: ParsCit: an Open-source CRF Reference String Parsing Package. LREC 2008

[8] [Jie et .al] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, Zhong Su:ArnetMiner: extraction and mining of academic social networks. KDD 2008: 990-998

[9] https://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html

[10] Abbas Nadine, Nasser Youssef and El Ahmad Karim. (2015). Recent advances on artificial intelligence and learning techniques in cognitive radio networks. Eurasip Journal on Wireless Communications and Networking. 2015. 10.1186/s13638-015-0381-7.

[11] Ho Tin Kam. Random decision forests. proceedings of the 3rd international conference on document analysis and recognition. 1995.

[12] term frequency-inverse document frequency
TFIDF, short for term frequencyâĂŞinverse document frequency

[13] Frequency Distribution of word in a document. frequency distribution for the outcomes of an experiment

[14] Natural Language Processing
Natural Language Processing

[15] pandas: powerful Python data analysis toolkit
pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

[16] Confusion matrix
evaluate the quality of the output of a classifier on the iris data set

[17] Word Net
A Lexical Database for English

[18] Mutinomial naive bayes
multinomial naive bayes sklearn

[19] Linear Support Vector Classification
A Lexical Database for English

[20] Decision tree
Decision tree

[21] Gaussian Naive Bayes (GaussianNB)
Gaussian Naive Bayes (GaussianNB)

[22] random forest classifier.
random forest classifier.