ACM Venue Recommendation System
CS6604 - Digital Libraries Final Project Report

Harinni Kodur Kumar and Tanya Tyagi

Virginia Polytechnic Institute and State University

Edward A. Fox, Instructor; Ziqian Song, GTA,

Dec. 23, 2019

Blacksburg, Virginia

Keywords: Recommendation, Classifiers, Conference Venues

ACM Venue Recommendation System
CS6604 - Digital Libraries Final Project Report

Harinni Kodur Kumar and Tanya Tyagi

(ABSTRACT)

A frequent goal of a researcher is to publish his/her work in appropriate conferences and journals. With a large number of options for venues in the microdomains of every research discipline, the issue of selecting suitable locations for publishing cannot be underestimated. Further, the venues diversify themselves in the form of workshops, symposiums, and challenges. Several publishers such as IEEE and Springer have recognized the need to address this issue and have developed journal recommenders. In the proposed project, the goal is to design and develop a similar recommendation system for ACM. The conventional approach to building such a recommendation system is to utilize the content features in a dataset through content-based and collaborative approaches, and proffer suggestions. An alternative is to view this recommendation problem from a classification perspective. With the success of deep learning classifiers in recent times and their pervasiveness in several domains, our goal is to solve the problem of recommending conference and journal venues by incorporating deep learning methodologies given some information about the submission like title, keywords, abstract, etc. The dataset used for the project is the ACM Digital Library metadata that includes metadata and textual information for research papers and journals submitted at various conferences and journals over the past 60 years.

Contents

List of Figures

List of Tables

List of Abbreviations

ACM Association for Computing Machinery

ANN Artificial Neural Networks

CCS Computing Classification System

CNN Convolutional Neural Networks

CSV Comma Separated File

DL Digital Library

JCDL Joint Conference on Digital Library

JSON JavaScript Object Notation

SIG Special Interest Group

SQL Structured Query Language

Chapter 1

Introduction

Research activities in an academic environment include publishing papers, citing related works, and collaborating with peers. Each of these activities involves making some critical decision that determines the outcome. Choosing a suitable research domain, deciding whom to collaborate with, determining the best venue for publishing research works, etc., are some of the decisions that have to be made with careful consideration. In the proposed work, we explore the problem of identifying the best venue to submit research products like papers, articles, etc. The problem could also be rephrased as a "potential venue matching problem". A venue could refer to the best conference or the best possible journal.

The motivation for the need to address the problem stems from the fact that there has been tremendous growth in the number of academic venues in recent times. Choosing an appropriate venue not only increases the probability of a work receiving recognition but also enhances the possibility of obtaining good reviews that can help a researcher refine his work. The number of ACM conferences as of date is at 120 as per [1]. This number is in addition to the Special Interest Group (SIG) conferences conducted by ACM. The number of possibilities within a particular computing discipline is high as well. For example, for the domain of information retrieval [20], many conferences exist, like SIGIR, CIKM, WSDM, WWW, and JCDL. A search on the IEEE website [2] would provide a huge list of conferences organized by that association. Thus, determining the appropriate set of venues for a work becomes all the more significant.

The problem can be viewed from the perspective of information overload as well. Whenever the number of plausible choices is huge, offering recommendations can help alleviate the burden of decision-making. Well known corporations, such as Amazon [19] and Netflix [11], are recognized for the personalized product recommendations they proffer, which in turn enhance customer satisfaction and revenue. The idea is to extend the approach of recommendation to address the information abundance in academia as well.

The venue matching problem is well-explored as demonstrated in [15, 16, 20]. Further, publishers like Elsevier [13] have a system recommending relevant journals to submit articles. Several other recognized publishing houses have recommender systems for submissions, like Springer [4], IEEE [18], and Wiley [5]. However, such a recommendation system has not been developed for ACM. In the proposed project, we aim to address that issue by adopting a deep learning-based classification approach to provide recommendations.

The popularity of deep learning in recent times cannot be understated. Application of deep learning in the fields of Computer Vision and Natural Language Processing (NLP) is burgeoning. The performance of traditional recommendation approaches built on the foundations of content-based filtering, collaborative filtering, or a hybrid mixture have considerable reliance on the quality of the features used. Extensive feature engineering is an expenditure that has to be borne for quality recommendations. However not every dataset can be attributed to having explicit features that help with building good predictive models. Deep learning ameliorates this issue with its excellent property of learning features from scratch, thereby making it desirable and effective for recommendations [22].

Chapter 2

Review of Literature

Recommendation systems have grown from being an additional feature in applications to an expected feature in most domains today. Sports, News, and Movies are the popular use-cases of recommendation systems. An extension of that idea is using recommendation systems in the domain of academia.

A considerable amount of work has been done in the past concerning suggesting research papers, collaborators, and research topics. [8] has demonstrated that over 80 different approaches have been used in the recommendation of academic literature. However, it is to be noted that of those works evaluated in this paper, very few have considered venue recommendation as a problem. One such work is [21], where a venue recommender is designed based on collaborative filtering approaches. The paper uses a combination of stylometric features and nearest neighbor papers to recommend venues based on CiteSeer data. Stylometric features of content in papers represent the number of words, the total number of sections, the total number of figures, etc. If a paper has a common author with another paper, the two papers are considered as neighbors. The authors use both metadata as well as full-text information about the submissions to recommend venues.

In a similar fashion, works like [16] use an author's publication network and the paper's content to predict venues. This work was tested using Special Interest Group (SIG) conference data of ACM only. Similarly, citation networks of the particular submission Alshareef et al., [7] have been explored well to make predictions. One such example would be [15], where

apart from just providing suggestions, relevant feedback mechanisms have been employed to solve improve the results of the suggestions. In [9], the co-authors, co-citers, and co-affiliated information are used to make the recommendations more personalized.

The works mentioned so far have used full text and citation information from a submission in addition to metadata information on different datasets. In [17], methods to offer preliminary recommendations using the title and abstract of a submission have been proposed. A Cavnar Trenkle based n gram (character gram) method has been used to create a language profile for each submission in the corpus. During the actual recommending phase, a language profile for the article is created as well. Finally, articles with the shortest distance to the test article are recommended to users. In addition, the authors have also used Latent Dirichlet Allocation to identify topics of venues and test articles. Nearest venues based on Euclidean distance are recommended to the users.

In terms of actual methods used, several approaches like topic analysis using Latent Dirichlet Allocation, collaborative filtering using features extracted from the content, and nearest neighbor classifiers have all been explored. [21] used matrix/tensor recommendation as well. In a similar vein, the content-based filtering approach has been used to recommend venues, as demonstrated in [10].

Another take on this problem has been to suggest a venue using a user's current research pursuit and interests as described in [6]. A researcher's scholarly reading behavior is used as a means of implicit feedback and to avoid cold start problems. Thus it can be observed that recommending venues has been explored from a traditional recommendation approach using content and collaborative approaches.

Another take would be to model this problem as a text classification problem. Text classification is the process of classifying a given text into predefined categories. We will be taking

this approach in the current paper. In the present context, the text refers to the information obtained from the user about a submission like a title, keywords, etc. The set of possible predefined categories or output classes is the set of conference venues and journals gleaned from the dataset.

In formulating recommending venues as a classification problem, a deep learning approach has not been widely explored. The closest work where a deep learning approach has been used in recommending for academia setting is in [20]. Deep artificial neural networks have been used to suggest venues for hosting meetup events.

Based on our study of the related work, there were a few issues which were common across developing academic recommender systems. Each system has been developed for a particular dataset. This makes the task of comparison of results between systems a challenging one. The next step is to develop a set of evaluation strategies to validate the results. Here we use the actual venue in which a work was published as a "gold standard," but note that other venues might also be appropriate, so point out that this matters adds complexity to the evaluation.

In the proposed work, our goal is to use a novel deep learning based classification method to recommend venues by making use of the title, keywords, CCS concepts, and abstract. Since, to the best of our knowledge, a venue recommendation has not been done for the ACM Digital Library (ACM-DL) dataset, we describe our open source solution to this need.

Chapter 3

Methodology

The ultimate goal of the project is to recommend to a user appropriate venues for a submission, given some information like title, abstract, etc. Like any typical deep learning project, the project pipeline is 1) Data Cleaning and Pre-processing, 2) Feature Extraction, 3) Model training, and 4) Evaluation.

3.1 Data cleaning and pre-processing

The ACM-DL data is in the form of XML files. Through this phase, data will be transformed to a format that can be fed to the models. A detailed description of data pre-processing and the work done will be discussed in Chapter 5. Currently, we have data corresponding to conference and journal specific records. Each record in the CSV file (obtained after our first level of pre-processing) represents a specific edition of a journal or conference. The next logical step in pre-processing is to use these records and build a database of entries that are submission or journal article specific. This would complete this stage by the end of which we would have data in the format of <input features, target label>.

3.2 Feature extraction and model training

After pre-processing, the problem boils down to a multi-class text classification problem as described in Chapter 2. The high-level idea of our plan is to use deep classifiers like Convolutional Neural Networks (CNNs) to classify the data into the closest conference venue or journal.

A deep neural network is a class of artificial neural networks (ANN) with more than one hidden layer between input and output layers. A CNN is a category of deep neural networks that involves a mathematical operation called convolution in place of traditional matrix multiplications in ANN ([3]). Convolution is a linear operation on two functions of a real-valued argument. Initially applied in the field of Computer Vision, CNNs have been shown to have promising results in the field of NLP.

In [14], sentences are classified by CNNs which can identify patterns in a text irrespective of location. The model is trained with a layer of convolution on top of word vectors obtained from an unsupervised word model called Word2Vec. In the proposed project, the initial features from a submission like a title and keywords will be converted to similar such word embeddings on which the CNN model will be trained to predict the classes.

As there are multiple conference venues, we plan to build multiple binary classifiers that are specific to each particular conference or journal. Later these could be combined to simulate a multi-class classifier. Although the plan is to utilize CNN to achieve the purpose, we also will experiment with different deep classifiers (like Recurrent Neural Networks) and architectures to identify the model that predicts the classes accurately. We aim to use the Python Keras framework to train and test our model.

3.3 Evaluation

The core evaluation strategy is to use accuracy and F score metrics to quantify how well the model classifies the data. We have not identified any such work on ACM data due to which we essentially do not have a baseline to compare with. A comparison against similar such works would imply comparing against different datasets like CiteSeerX, IEEE, DBLP, etc. The actual conferences and journals where the submissions appeared will be used as ground truth labels.

3.4 Architecture of the system

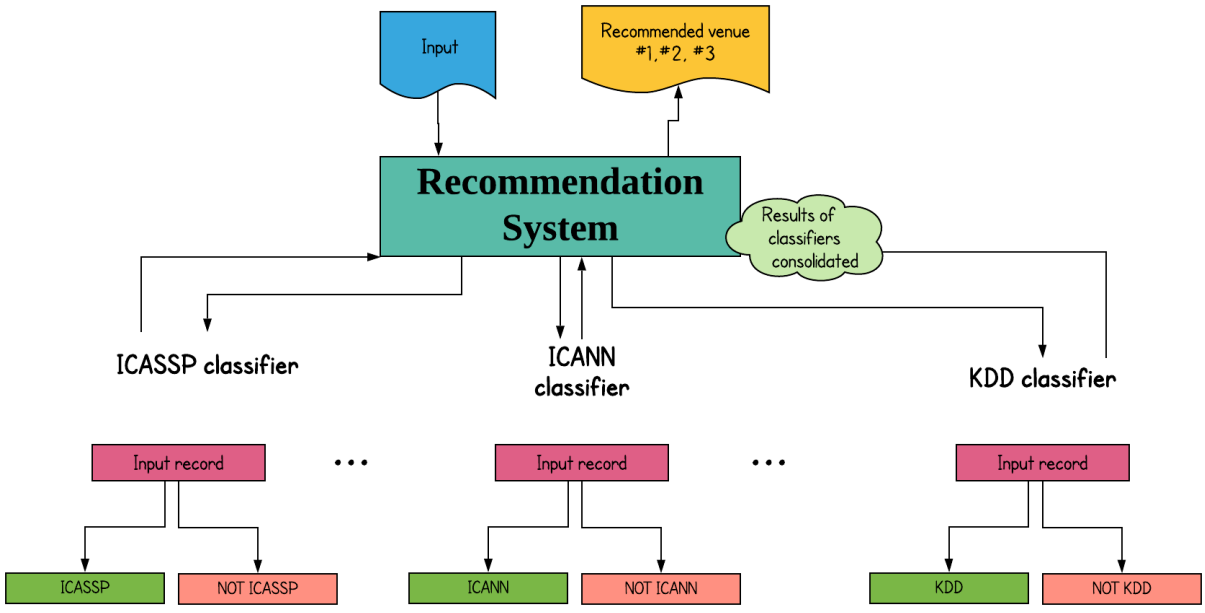


Figure 3.1: Architecture of the system

Figure 3.1 represents the current system architecture. An input record with necessary information is passed to several binary conference/journal classifiers. Each classifier in turn would return an output probability of whether a record belongs to a particular class or not. Such information is consolidated from all the classifiers, ranked and a final set of venues is recommended.

Chapter 4

Dataset pre-processing

The Association for Computing Machinery - Digital Library (ACM-DL) staff has provided us with metadata information about the conferences and journals for a period of more than 60 years. The metadata contains information published at non-ACM conferences as well. The data is in the XML format with about 20,150 files about conferences and 73,253 files about journals. The subsequent sections provide a detailed discussion of the data.

4.1 Data organization

The ACM dataset has two parts: proceedings (conferences) and periodicals (journals).

4.1.1 Proceedings

Each file in the proceedings corpus describes a single conference. The XML file could be divided into two logical sections: 1) metadata section and 2) content section. The metadata section defines the properties of the conference under consideration. Although this section has several tags representing different features of the conference, the important fields which will be useful for the system are listed below. For this report, the term XML tags will be used interchangeably with the term fields.

- conference rec
  - start date
  - end date
  - city
- proceeding rec
  - id
  - acronym
  - proc_desc
  - proc _title
  - proc _class
  - publication date
- publisher
  - ID
  - code
  - name
- categories

These metadata fields about the conference contain text that helps in formulating the target labels for the recommendation. As pointed out earlier, in the first iteration the problem of recommending conferences or journals is treated as a classification problem. Every classification system is characterized by input features and output labels. The fields listed above are used to create the output labels.

The output labels for the proposed system are the conference names. The conference name is obtained by concatenating information from fields like *proc_title*, *proc_desc*, and *proc_class*. In a similar fashion, each file represents information about a specific edition of a specific conference. Following is a sample representation:

<proc>2226954</proc>

<acronym>1LeGE-WG'02</acronym>

<proc_desc>Proceedings of the 1st LEGE-WG international conference</proc _desc>

<conference _number>1</conference _number>

<proc _class>workshop</proc _class>

<proc _title>Educational Models for GRID Based Services</proc _title>

From the above XML snippet, it can be observed that the conference is called *"Proceedings of the 1st LEGE-WG international conference on Educational Models for GRID Based Services"*. Here, the conference name is obtained by concatenating information from the *proc_desc* and *proc _title* fields. It is common for conference proceedings to be addressed in short using specific acronyms. The conference acronym for the above example is *"LeGE-WG"*. Both the acronym and the conference names serve as important fields for the current project.

The content sections describe the official work submitted and accepted at a particular conference. A single conference in a particular year can have submissions ranging from 10 papers to about 50 papers. A few of the important fields from the content section are listed below.

- article _
  - article _id
  - article publication date

- title
- ccs
- abstract
- keywords
- authors
- references

Each submission or article accepted in the conference is marked by the *article* tag. This would mean the content section of the conference file can contain up to 50 article tags indicating 50 submissions. Each article is further identified by an ID, date of publication, etc. Most of the fields are self-explanatory. One field of importance is the CCS concepts field.

ACM 2012 Computing Classification System (CCS) is a widely used method to classify submissions at ACM conferences. The goal of the system is to tag each submission with appropriate categories which can be useful for authors, reviewers, etc. Further, classifying submissions in this established format can be useful for indexing the ACM-DL as well. Each CCS term refers to a concept in a computing discipline. The system is a six-level poly-hierarchical structure. Figure 4.1 depicts this structure.

The CCS concepts as identified by the CCS field of an article will serve as an important feature in classifying the submissions. Further, the content section is useful in creating the input features for the classification problem. Given an article with a certain title, keywords, CCS concepts, and abstracts, our goal would be to predict an appropriate venue.

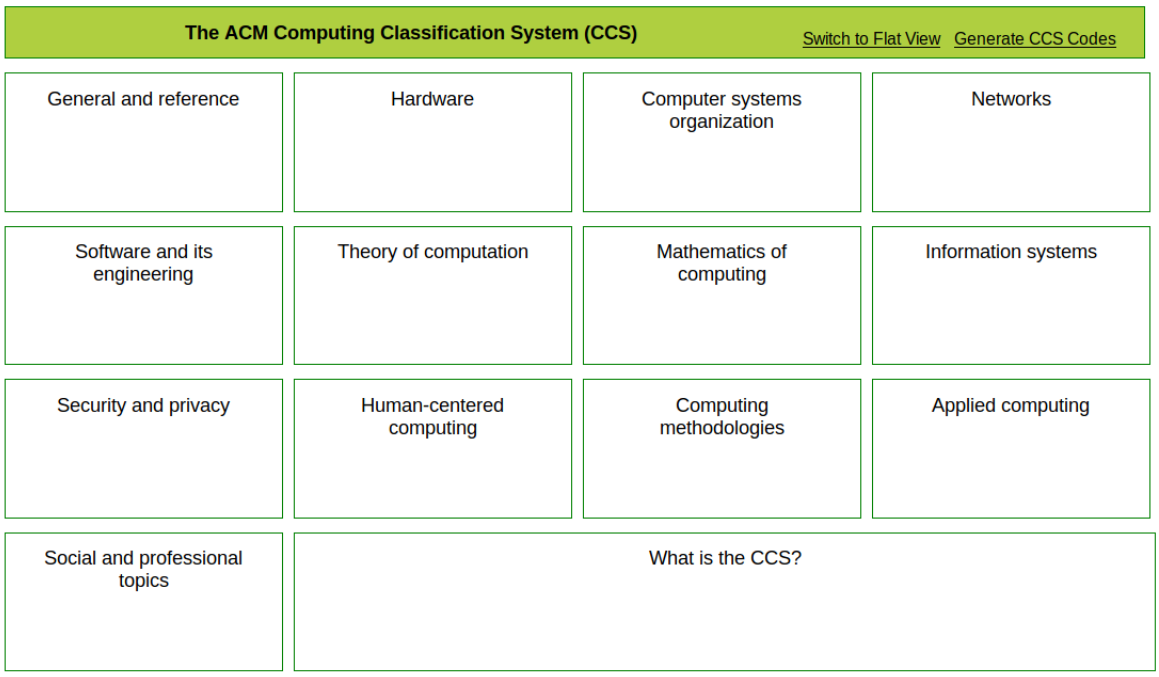


Figure 4.1: Categories in the ACM classification system

4.1.2 Periodicals

The second category of the records have information about the various journals. Similar to conference records, each journal XML file describes a specific edition of a specific journal section with metadata and content sections. Important fields from the metadata section are listed below.

- journal _id
- journal code
- journal name
- publisher
  - id

  - code
  - city

The journal name field will serve as an output label for the journal classification. Journal code is the acronym for a specific journal. The content section of the journal has the following important fields.

- article _id
  - article _id
  - article publication date
  - title
  - url

4.2 Data pre-processing

Data pre-processing is the initial stage of any machine learning or deep learning project pipeline. The process aims to transform the data into a format that can be fed into the models. So far the challenging aspect of the project has been the data pre-processing phase. The key steps in the pre-processing phase are data chunking and data cleaning.

4.2.1 Data chunking

The data received from ACM in the form of XML files are of size 9 GB and 11 GB for the periodicals and proceedings folders, respectively. Loading files from the two folders was a

time-consuming process. To aid with the idea of developing a quick prototype and to make working with data files easier, the two folders were chunked into several smaller folders.

4.2.2 Dataset creation

The metadata dispersed in XML files is converted to CSV format. The conference CSV file has the fields

<Sno, Proc Id, Acronym, Year, Proceedings Title, Publication Date, Publisher>

Similarly, the Journal CSV file has the fields **<Sno, Code,Name, Type, Issue Date>**

The organization of data in this format is to facilitate the cleaning of these important fields. With these files, the subsequent step of data cleaning is performed. During the process of performing this step, we realized transforming data into JSON format might be more useful as JSON files are more convenient to work with.

4.2.3 Data cleaning

One of the key challenges has been in establishing a consistent name for the conferences. To cite an example for this, *"PAM"* is a conference on network measurement and analysis. In the initial edition, the conference was termed "passive and active measurement," but then it was changed to "passive and active network measurement". Similarly, the "APNOMS" conference has the names "Asia-Pacific Network Operations and Management enabling the future internet for changing business" and "Asia-Pacific Network Operations and Management" used interchangeably in the dataset. "SIGCSE" conference proceedings have the title "Technical Symposium on Computer Science Education". Some editions have the the publisher information "ACM" and the conference acronym "SIGCSE" in the title as well. Another example of inconsistency would be the inclusion of the year of the conference in the

title field. Despite having a separate year field in the metadata of the conference, several conferences have year information in the title as well.

Several conferences have such minor modifications in their names over the years. Although having such minor differences in the conference titles does not affect a human, it will make a difference when building a recommendation system. The additional words or lack of them in the titles which serve as the target labels might lead the classifier to make incorrect assumptions. Thus, converting the conference titles or venue names to standard format is an important data cleaning step.

To present an accurate picture, a complete conference proceeding title would be as follows *"Proceedings of the 2006 international conference on Software Process Simulation and Modeling"*. It could be seen that "Software Process Simulation and Modeling" is a piece of important information in the title which describes the conference in a few words. The first part of the title might not be that useful to a classifier in predicting since that information will be present in almost every conference. Besides, year can be obtained from a different field. During the data cleaning stage, such terms are removed. Such redundant terms are identified and we call them **conference stopwords**. The list includes terms like 1) Proceedings of the, 2) Conference, 3) Workshop, 4) Challenges, and 5) Ordinals indicating the edition of a conference like "first", "second" etc., 5) Well known publisher names like IEEE and ACM.

Once these conference stop words are removed, we are mostly left with the meat of the conference title. Similarly, year information like "2006" or variations of it like "06" and "6," are removed from the title. Thus, as a result of this data cleaning process, a title of *"Proceedings of the 2006 international conference on Software Process Simulation and Modeling"* would be converted to *"Software Process Simulation and Modeling"*. Table 4.1 gives additional examples.

Table 4.1: Examples for conference acronym to venue mapping

Conf Acronym	**Venue terms**
CAMRa	context-aware movie recommendation
SPW/ProSim	software process simulation and modeling
EvoCOP	European evolutionary computation in combinatorial optimization
MPRSS	multimodal pattern recognition of social signals in human-computer-interaction
TAMODIA	task models and diagrams for users interface design
OOPSLA	addendum to the object-oriented programming systems...
ASWEC	australian software engineering
EUROSEC	european system security
WONS	wireless -demand network systems and services
ISIE	intelligence science and information engineering
WSNA	wireless sensor networks and applications

4.2.4 Submission specific database

The dataset created above is conference/journal specific. However, the models will work on features that are specific to a submission. To help with that goal, a database of papers in conferences and journals has been constructed. Each row in the database table corresponds to a specific submission in a conference/journal and contains information relevant to that.

The dataset was converted as follows. Every XML file contained information for about 10 to 50 papers. With the help of the Python Beautiful Soup library, the fields of articleid, title, abstract, authors, full-text, keywords, CCS concepts, name of the proceedings, AND conference acronym were extracted for each of those papers in a conference. A mapping from the proceedings name to the cleaned venue name was stored in a Python dictionary. The proceedings information extracted from the paper is compared against this dictionary to obtain the corresponding venue information. The original XML file contains additional information about fields like authors, keywords, categories, etc. For example, we can find information like author location, designation, place of work, etc. Similarly, the level of CCS concept information is also available. However, for our project, we discard that information

and extract only the essential content. The following fields are parsed, cleaned, serialized into JSON format, and then stored in the database. Figure 4.2 shows a sample file represented in JSON format.


```
{
    "sno": "1",
    "article_id": "2390778",
    "title": "Intelligent menu planning",
    "abstract": "\n<p>With the growth of recipe sharing services, online cooking recipes associated with ingredients a
sharing sites have ...",
    "fulltext": "\n Intelligent Menu Planning: Recommending Set of Recipes by Ingredients Fang-Fei Kuo Department of (
    "date": "11-02-2012",
    "keywords": [
        "cooking",
        "ingredient",
        "menu planning",
        "minimum steiner tree",
        "recipe recommendation"
    ],
    "ccs": [
        "Retrieval tasks and goals",
        "Data mining",
        "Document filtering",
        "Information retrieval",
        "Information systems applications",
        "Information extraction",
        "Information systems"
    ],
    "category": [
        "Data mining",
        "Information filtering"
    ],
    "conference": "Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities",
    "venue": "multimedia multimedia for cooking and eating activities",
    "acronym": "CEA"
}
```


Figure 4.2: Sample paper data converted into JSON format

sno, entry_id, title, kw, CCS, abstract, fulltext, conference, venue, authors

Database schema for journal and conferences are provided in Figure 4.2 and Figure 4.3.

The sno fields in both the tables are the primary keys. The entryid field is the ID of the article in the original XML files. The field category helps classify files in one of the many general Computer Science categories. KW represents the keywords found in the papers. Similarly, terms in journal data represent the high level ideas of an article. Entry_date is the date of submission. acrn represents an acronym of the conference or journal name.

An example entry for conference data is shown in Figure 4.4. A total of around 1 million submissions were available for all the conferences.

Table 4.2: Conference data schema

Field	**Type**
sno	varchar(255)
entry_id	varchar(255)
title	text
ccs	text
kw	text
category	text
abstract	longtext
ftext	longtext
conference	text
venue	text
acrn	varchar(255)
entry_date	date
authors	text

Table 4.3: Journal data schema

Field	**Type**
sno	varchar(255)
entry_id	varchar(255)
title	text
subtitle	text
authors	text
ccs	text
terms	text
category	text
abstract	longtext
ftext	longtext
journal type	text
journal	text
venue	text
acrn	varchar(255)
entry_date	date

4.3 Analysis of the dataset

We describe the results of our study on the data. The section has been divided into proceedings and periodicals sections for ease of reading.

Table 4.4: Format of the data

sno	1000
entry_id	1732691
title	Finding and exploiting goal opportunities in real-time during plan execution
ccs	NULL
kw	NULL
category	NULL
abstract	Autonomous robots that operate in real-world...
ftext	NULL
conference	Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots...
venue	eee/rsj intelligent robots and systems
entry_date	2009-10-10
authors	Paul Schermerhorn

4.3.1 Proceedings

The received data has information about conference submissions from 1951 till 2014. Data distribution of submissions over the years is shown in Figure 4.3

From the figure, the gradual increase in the number of submissions across the years could be seen, with 2010 having the highest number of submissions. Among all the submissions from 1951, a total of 4786 unique conferences were found with a combined total of 20,152. This number includes the conferences and their editions over the years. Of these, 1935 conferences did not have an acronym associated with it. Such conferences were mostly before the year of 1985 and hence will not be used in the project.

Each conference is associated with a particular publisher. Although the data is obtained from ACM, the corpus contains submissions published by others such as IEEE and Springer. Table 4.5 provides a list of the top 5 publishers, with the most submissions across years.

It can be observed that out of all the conferences, *SIGGRAPH*, which stands for *Special Interest Group in Computer Graphics*, has the most editions, at 148 over the years. These

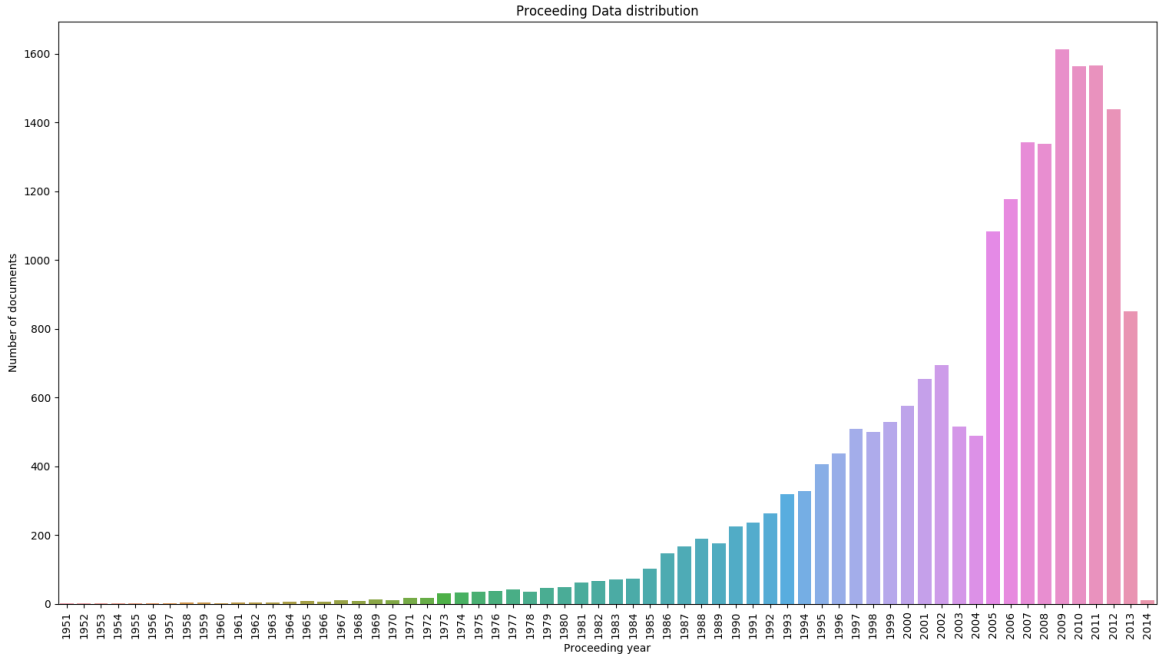


Figure 4.3: Proceedings submissions vs. Years

Table 4.5: Publishers with most conference submissions

Sno	Publisher	Submission count
1	Springer-Verlag	6584
2	Association for Computing Machinery (ACM)	5122
3	Institute of Electrical and Electronics Engineers Computer Society (IEEE)	4732
4	Association for Computational Linguistics (ACL)	519
5	World Scientific and Engineering Academy and Society (WSEAS)	359

editions include the annual conferences, workshops, challenges, and related events under the common name of SIGGRAPH. This is followed by *HICSS - Hawaii International Conference on System Sciences* at 108 editions. It is interesting to note that the first conference of HICSS took place in 1968, which is earlier than SIGGRAPH, which was inaugurated in 1974. However, to ensure the recommendations are relevant to 2019, data from 2000 to 2014 will be used for recommendation.

Table 4.6: Publishers with most journal submissions

Sno	Publisher	Issues
1	Association for Computing Machinery	9001
2	Elsevier Science Publishers B. V.	8293
3	Kluwer Academic Publishers	6040
4	IEEE Press	4989
5	Academic Press Inc.	3022

Table 4.7: Journals with most issues

Journal	Issues Count
Theoretical Computer Science	823
Communications of the ACM	677
Information Processing Letters	598
ACM SIGPLAN Notices	565
IEEE Transactions on Computers	552
Fuzzy Sets and Systems	550
The Computer Journal	484
Journal of Computational and Applied Mathematics	480
Fundamenta Informaticae	466
Journal of Computational Physics	450

4.3.2 Periodicals

A total of 73,253 journals were identified in the ACM-DL data. Of these, 1334 unique journals were obtained. A total of 220 journals have more than 100 issues since 1956. Among these unique journals, *TCSC*, which is the shortened version of *Theoretical Computer Science*, has the highest number of editions at 823.

Table 4.6 lists the top publishers of the journals in print over a long period of time.

Based on the existing information, we identified the journals with the highest number of issues. Information about the top 10 journals appears in Table 4.7. The journal data has further been subdivided into 1) Journal (63621), 2) Magazine (1409), 3) Newsletter (5053), and 4) Transaction (2130). The numbers in the parentheticals indicate the total number of

issues in that category. Like with conferences, the project data from 2000 to 2014 will be used. This encompassed a total of 43,689 journal files. Each file represents a particular issue of a specific journal.

Chapter 5

Models and Evaluation

The next step in the process is the building of binary classifiers for each of the conferences. Currently, we have been testing binary classifiers for 50 conferences and 30 journals. We experimented with logistic regression classifier initially. Given the corresponding input and labels, logistic regression had a training accuracy of about 0.98 and a testing accuracy of 0.95 for 5 classifiers. However it did not perform well as the number of classifiers increased. So we decided not to proceed with it.

The general methodology is as follows. Consider a sample binary classification problem of whether a record can be classified as belonging to the SIGCSE conference. The data is extracted from the database with a mix of data from *SIGCSE* as class 1, and data from other conferences as class 2. For the considered example, the number of records in the SIGCSE class was about 100 and about 100 records from other conferences were chosen. The fields title, abstract, CCS, and keywords were extracted for a record and concatenated into a single field. The single textual field is count vectorized to obtain the frequency of each term. This count vectorized textual data becomes the input features for the model. The acronym field from the database that represents the abbreviated form of conference name becomes the target label for our model. The data was split into training and test sets with a 80-20 ratio.

5.1 A brief overview of the models used

We built Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) based classifiers.

ANNs cover a large number of neural network architectures, such as CCNs, LSTMs, Auto Encoders, etc. In general, ANNs are a collection of artificial neurons/nodes. The different ways in which the layers of individual units are connected to each other forms different architectures. CNNs, as will be discussed in detail next, are a specific type of ANN that have one or more fully connected layers, called convolutional layers.

CNNs are a type of neural network that can process data that has a grid like topology ([12]). They use a convolution operation in place of matrix multiplication in at least one of their layers. CNNs have had considerable popularity in image classification.

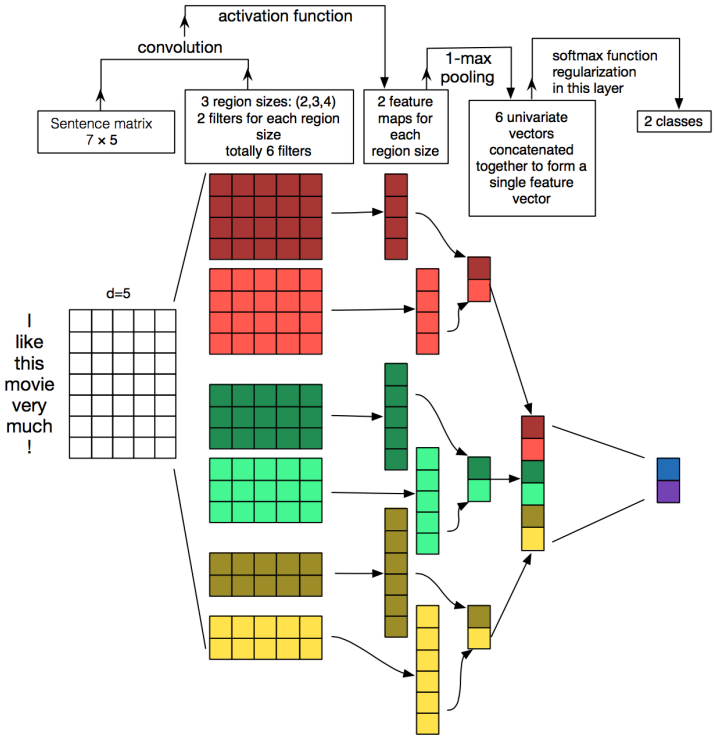


Figure 5.1: Text classification with CNN

The important aspect of CNNs has been their ability to capture spatial and temporal features of an input dataset with the help of filters. In the last few years, works have been focused on utilizing CNNs for text classification [23]. An architecture taken from the paper, best describes how CNNs can help in text classification as shown in Figure 5.1.

5.2 Artificial Neural Networks

Artificial neural networks, when executed with 100 epochs, had 100% training and testing accuracy. When trained over 40 epochs, the training accuracy was betwee 1 and 0.95. The predictions and accuracy obtained at this stage appear to overfit the training data.A

ccuracies around the same range were obtained for other binary classifiers like JCDL, CEA, etc. Since the overfitting issue could not be resolved, we went ahead with our next model.

5.3 Convolutional Neural Networks

The intuition is to build a binary classifier for every conference. So when a recommendation is to be made for a submission, the request is passed to all the classifiers. The results of all the classifiers are combined and ranked to make a recommendation. With that high level idea, several binary CNN classifiers have been modelled. The input features to the CNN models is the same as that for ANN as described in the previous section.

The architecture of the CNN model is simple, with an embedding layer, 1D convolution layer, a max pooling layer, one fully connected layer with 10 hidden units and an output layer that predicts 2 classes. RELU and sigmoid activation functions have been used. Also we use an Adam optimiser and binary cross entropy loss functions.

The number of data points in positive and negative classes depend on the classifier under consideration. For example, for a binary classifier modelled for the *WWW* conference, the positive class (i.e., records with class as WWW) has 2178 records. For the negative sample, we chose around 300 records from each of 50 other chosen classifiers. These 50 classifiers are the conferences with the largest number of submissions between the period of 2000 and 2014. Similarly binary classifiers were built for about 30 journals. The conference/journal names of these classifiers can found in Appendix C.

Table 5.1 lists the number of positive samples chosen for a few of those classifiers. The negative samples are kept constant at 6900. The classifier name represents the shortened

version of the conference name.

Table 5.1: Class distribution

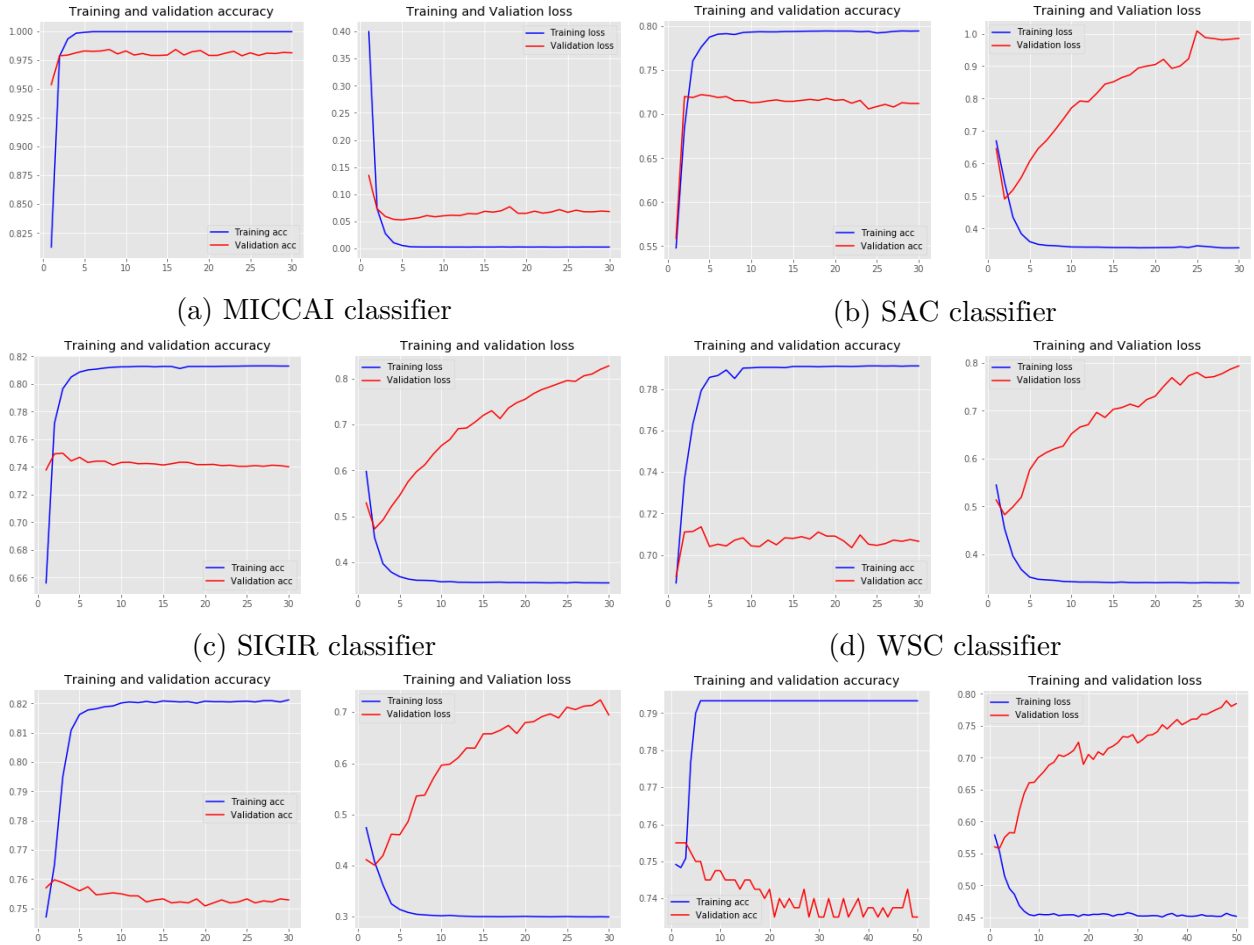
Classifier	Positive Samples
AAMAS	3634
CHI	2852
CIKM	3379
CVPR	4884
DATE	3728
DEXA	2738
FSKD	3651
SIGCSE	2152
SIGIR	2409
WWW	2178
ICPR	4884

5.4 Evaluation

In addition to the training, validation, and test sets for each of these classifiers, about 200 data samples from each of these conferences and journals were set aside to form the *unseen data*. The performance of all these classifiers were tested using that data. This translates to having 200 samples in positive class and the rest for the negative class.

The actual venues found in the dataset were treated as ground truth labels. The metrics used for testing the performance of the classifiers were: 1) Accuracy, 2) F score, 3) Was ground truth recommended?, 4) Were similar venues recommended?

Most of these classifiers had training and validation accuracy around 0.8 and 0.7, respectively. The training and validation accuracy and their losses can be seen as shown in Figure 5.2.



(a) MICCAI classifier

(b) SAC classifier

(c) SIGIR classifier

(d) WSC classifier

(e) WWW classifier

(f) SIGCSE classifier

Figure 5.2: Accuracy and Loss for six sample classifiers

Given the imbalanced nature of the unseen data, the testing accuracy was high in most of the cases. As a result an F score might be a good estimator of the performance. F scores of a few of the classifiers are presented in Table 5.2.

In Table 5.2, columns 1 to 6 represent conference values, and the rest represent the journals. The high F score for the negative class is a sign of the imbalanced data set. In general, we observed that the F scores for the journals were better than for the conferences. This could be attributed to the fact that most of the journals do not have missing data in the input

Table 5.2: F scores for the classifiers on unseen data

Sno	Classifier	F score +ve	F score -ve
1	ICASSP	0.81	0.98
2	SODA	0.80	0.91
3	ECCV	0.49	0.96
4	ICANN	0.32	0.90
5	KES	0.28	0.90
6	AINA	0.27	0.93
7	ITCO	0.70	0.93
8	IPRL	0.60	0.92
9	SIGN	0.81	0.95
10	FSTS	0.86	0.97
11	JCAM	0.47	0.96
12	JOCP	0.83	0.95
13	IEEENETW	0.63	0.98
14	IJNM	0.72	0.97
15	TCSC	0.58	0.91
16	FUNI	0.35	0.97
17	COMS	0.33	0.97
18	SPRE	0.71	0.91
19	PTRL	0.53	0.98
20	ITHR	0.68	0.98

Table 5.3: Classifier predictions

Input	Actual label	Predicted label
a semiblind em algorithm for overcomplete independent component analysis	ICASSP	ICASSP
performance measurement of data transfer services	IEEENETW	IEENETW
a lower bound on complexity of optimization on the wiener space	TCSC	TCSC
asymptotic analysis of a computational method for time and frequency dependent radiative transfer	JOCP	JOCP
an alternate relaxation approximation to conservation laws	JCAM	JCAM

feature fields. Conferences have missing data in CCS fields particularly if they are from the years between 2000 and 2005. Conferences, in particular the SIG conferences, have lower F scores. We are experimenting with other features to check if they improve our F scores. A Few of the predictions made by the binary classifiers are presented in Table .

5.4.1 Recommendations

The top 3 recommendation for a few of the test records are presented in Table 5.4. The input column in the table is the title of the submission. However the actual input to the model is a consolidation of title, CCS, abstract, and key words. GT represents the ground

Table 5.4: Recommendations

Input	1st rec.	2nd rec.	3rd rec.
On covariance function tests used in system identification	AJIF	JACM	ISIC
Efficient processing of topk twig queries over probabilistic xml data	WWWJ	FUNI	TOIS
Noise subspace fuzzy C means clustering for robust speech recognition	ICASSP *GT =ICCSA*	SIGN	-
Designing the model human cochleaan ambient crossmodal audiotactile display	IEEE-H	-	-
very sparse random projections	ITIOS *GT =KDD*	SIGIR	ISCI

truth label. In two examples, ground truth was not recommended. The correct conference is indicated in braces.

In several cases, the ground truth label was suggested as the first recommendation, as can be seen from the first three examples. For the last two records, even though the ground truth label of KDD and ICASSP was not recommended, the recommended journals were similar to these ground truth conferences. For example, ITIOS, SIGIR, and ISCI are information science related journals.

The one pressing issue that we noticed was the lack of recommendation in certain cases. For example, in our current subset of 80 classifiers, we do not have a journal that is similar to the IEEE Haptics conference. As a result, no similar venues are suggested. This leads to the bigger question of what recommendation to offer when we do not have a similar venue. The second major issue is the lack of recommendation of similar venues. To cite an example, consider the case of the input text *medium access control of wireless lans for mobile computing* . The submission with this title was published in the IEEE NETW journal. The journals of IJNM should be recommended too, as it is about computer networks as well. In our cases, the ground truth label is predicted correctly. However, a similar venue is not recommended.

5.5 Challenges

Some issues like low F scores and lack of similar venue recommendations were explained in the previous section. One of the ways to deal with similar venues would be to perform a topic analysis on the abstract and venue names and add that as a feature. We aim to do that in the next iteration.

ACM conferences in the later years have CCS concepts. However submissions from other publishers do not have them, leading to a missing data case. A topic analysis would help deal with this issue as well. Further, adding more classifiers could help deal with the lack of recommendation for some of the cases.

Bibliography

[1] Association for Computing Machinery Conferences. URL https://dlnext.acm.org/conferences. [date accessed Dec 23, 2019].

[2] IEEE Conferences. URL https://cis.ieee.org/conferences/conference-calendar. [date accessed Dec 23, 2019].

[3] Deep learning book. URL http://www.deeplearningbook.org/contents/convnets.html. [date accessed Dec 23, 2019].

[4] Springer Journals. URL https://journalsuggester.springer.com/. [date accessed Dec 23, 2019].

[5] Wiley Journals. URL https://authorservices.wiley.com/author-resources/Journal-Authors/find-a-journal/index.html. [date accessed Dec 23, 2019].

[6] Hamed Alhoori and Richard Furuta. Recommendation of scholarly venues based on dynamic user interests. *Journal of Informetrics*, 11(2):553 – 563, 2017. ISSN 1751-1577. doi: https://doi.org/10.1016/j.joi.2017.03.006. URL http://www.sciencedirect.com/science/article/pii/S1751157716303406.

[7] Abdulrhman Alshareef, Mohammed Alhamid, and Abdulmotaleb El Saddik. Academic venue recommendations based on similarity learning of an extended nearby citation network. *IEEE Access*, PP:1–1, 03 2019. doi: 10.1109/ACCESS.2019.2906106.

[8] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. Research paper recommender system evaluation: A quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and*

Replication in Recommender Systems Evaluation, RepSys '13, pages 15–22, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2465-6. URL http://doi.acm.org/10.1145/2532508.2532512.

[9] I. Boukhris and R. Ayachi. A novel personalized academic venue hybrid recommender. In *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 465–470, Nov 2014. doi: 10.1109/CINTI.2014.7028720.

[10] T. Ghosal, A. Chakraborty, R. Sonam, A. Ekbal, S. Saha, and P. Bhattacharyya. Incorporating full text and bibliographic features to improve scholarly journal recommendation. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 374–375, June 2019. doi: 10.1109/JCDL.2019.00077.

[11] Carlos A. Gomez-Uribe and Neil Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19, December 2015. ISSN 2158-656X. URL http://doi.acm.org/10.1145/2843948.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN 0262035618, 9780262035613.

[13] Ning Kang, Marius A. Doornenbal, and Robert J.A. Schijvenaars. Elsevier journal finder: Recommending journals for your paper. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 261–264, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3692-5. URL http://doi.acm.org/10.1145/2792838.2799663.

[14] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181.

[15] Onur Küçüktunç, Erik Saule, Kamer Kaya, and Ümit V. Çatalyürek. Recommendation on academic networks using direction aware citation analysis. *CoRR*, abs/1205.1143, 2012. URL http://arxiv.org/abs/1205.1143.

[16] Hiep Luong, Tin Huynh, Susan Gauch, Do Loc, and Kiem Hoang. Publication venue recommendation using author network's publication history. In *Intelligent Information and Database Systems*, volume 7198, pages 426–435, Mar 2012. doi: 10.1007/978-3-642-28493-9_45.

[17] Eric Medvet, Alberto Bartoli, and Giulio Piccinin. Publication venue recommendation based on paper abstract. In *IEEE 26th International Conference on Tools with Artificial Intelligence*, pages 1004–1010, Nov 2014. doi: 10.1109/ICTAI.2014.152.

[18] IEEE publication recommender. URL https://publication-recommender.ieee.org/home. [date accessed Dec 23, 2019].

[19] B. Smith and G. Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3):12–18, May 2017. doi: 10.1109/MIC.2017.72.

[20] Z. Yang and B. D. Davison. Venue recommendation: Submitting your paper with style. In *2012 11th International Conference on Machine Learning and Applications*, volume 1, pages 681–686, Dec 2012. doi: 10.1109/ICMLA.2012.127.

[21] Z. Yang, D. Yin, and B. D. Davison. Recommendation in academia: A joint multi-relational model. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 566–571, Aug 2014. doi: 10.1109/ASONAM.2014.6921643.

[22] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender

system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1):5:1–5:38, February 2019. ISSN 0360-0300. URL http://doi.acm.org/10.1145/3285029.

[23] Y. Zhang and B. Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taiwan, Nov 2017. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1026.

Appendices

Appendix A

Future work

1. Extend it for remaining conferences and journals
2. Model LSTM networks on the data
3. Do a topic analysis, add full text to the list of input features
4. Develop an interface for the recommendation

Appendix B

Python libraries used

Few of the libraries used are listed below.

- **BeautifulSoup**: Parsing xml files
- **json**: Python object to json creation
- **pymsql**: Connector to work with mysql
- **logging**: Logging features
- **pandas**: Data Manipulation
- **numpy**: Data Manipulation
- **path,os**: File handling libraries
- **keras, scikit**: Deep learning/ machine learning libraries
- **seaborn,matplotlib**:Data visualization

Appendix C

Names of the conference classifiers

We built 50 binary classifiers for conferences. Names of those conferences is listed here

- AAMAS: Autonomous Agents and Multiagent Systems)
- AINA: Advanced Information Networking and Applications
- ASPDAC: Asia and South Pacific Design Automation Conference
- CHI: Conference on Human Factors in Computing Systems
- CIKM: Conference on Information and Knowledge Management
- CIS: Conference on Computational Intelligence and Security
- CIT: Conference on Computer and Information Technology
- COMPSAC: Conference on Computers, Software and Applications
- CVPR: Computer Vision and Pattern Recognition
- DATE: Design Automation and TEst
- DEXA: Database and Expert Systems Applications
- ECCV: European Conference on Computer Vision
- FSKD: Fuzzy Systems and Knowledge Discovery

- GECCO: Genetic and Evolutionary Computation Conference
- HICSS: Hawaii International Conference on System Sciences
- ICANN: International Conference on Artificial Neural Networks
- ICASSP: International Conference on Acoustics, Speech and Signal Processing
- ICCAD: International Conference On Computer Aided Design
- ICCS: International Conference on Computational Science
- ICCSA: International Conference on Computational Science and Applications
- ICDAR: International Conference on Document Analysis and Recognition
- ICDE: International Conference on Data Engineering
- ICDM: International Conference on Data Mining
- ICIC: International conference on Intelligent Computing
- ICICIC: International Conference on Innovative Computing, Information and Control
- ICICTA: International Conference on Intelligent Computation Technology and Automation
- ICMTMA: International Conference on Measuring Technology and Mechatronics Automation
- ICNS: International Conference on Networking and Services
- ICPADS: International Conference on Parallel and Distributed Systems
- ICPR: International Conference on Pattern Recognition

- ICSE: International Conference on Software Engineering
- IIH-MSP: International Conference on Intelligent Information Hiding and Multimedia Signal Processing
- IJCAI: International Joint Conferences on Artificial Intelligence
- IPDPS: International Parallel and Distributed Processing Symposium
- ITNG: International Conference on Information Technology
- KDD: Knowledge Discovery and Data Mining
- KES: Knowledge-Based and Intelligent Information Engineering Systems
- LCN: Local Computer Networks Conference
- MICCAI: Medical Image Computing and Computer Assisted Intervention
- MM: Conference on Multimedia
- SAC: Symposium on Applied Computing
- SCC: Conference on Services Computing
- SIGCSE: Special Interest Group on Computer Science Education
- SIGIR: Special Interest Group on Information Retrieval
- SIGMOD: Special Interest Group on Management of Data
- SODA: Symposium on Discrete Algorithms
- VLSID: International Conference of VLSI Design
- WI-IAT: Web Intelligence and Intelligent Agent Technology

- WSC: Winter Simulation Conference
- WWW: World Wide Web Conference

List of journals for which the classifiers were built.

- AJCL: Computational Linguistics
- AJIF: Automatica Journal of IFac
- CACM: Communications of the ACM
- CCOMP: concurrency and COMPutation practice
- COMAG: COmmunications MAGazine
- COMP: COMPuter
- COMS: COMuter communicationS
- DAMA: Discrete Applied MAthematics
- FSTS: Fuzzy SeTs and Systems
- FUNI: FUNdamenta Informaticae
- IEEENETW: IEEE NETWorks
- IEETH: IEEE Haptics
- IJNM: International Journal on Network Management
- IPRL: Information PRocessing Letters
- ISCI: Information SCIences

- ISOF: IEEE transactions on SOFtware engineering
- ITCO: IEEE Transactions on COmputers
- ITHR: IEEE Transactions on information Theory
- JCAM: Journal of Computational and Applied Mathematics
- JOCP: Journal Of Computational Physics
- LINX: Linux journal
- PTRL: Pattern Recognition Letters
- SIGN: SIGNal processing
- SPRE: Software PRactice Experience
- TCSC: Theoretical Computer SCience
- TOCL: Transactions On Computational Logic
- TOIS: Transactions On Information Systems
- WIRE: WIREless networks
- WSTOIS: WSeas Transactions On Information Science
- WWWJ: World Wide Web