

VISUAL ANALYTICS OF RESEARCH COMMUNITY EXPERTISE
IN SPACE AND TIME

by

Deepak Munjal

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2019

© Copyright by Deepak Munjal, 2019

*To Mom (Ramesh Munjal), Dad (Late Ram Lal Munjal) and Brother
(Lakshay Munjal)*

Table of Contents

List of Tables	v
List of Figures	vi
Abstract	viii
List of Abbreviations Used	ix
Acknowledgements	x
Chapter 1 Introduction	1
Chapter 2 Related Work	7
2.1 Introduction	7
2.2 Data Collection	8
2.3 Disambiguation to Wikipedia	9
2.4 Visualization	9
Chapter 3 Methodology	11
3.1 Introduction	11
3.2 Document Representation	13
3.2.1 Bag of Words	13
3.2.2 Bag of Concepts	14
3.2.3 Bag of Categories	15
3.3 Classification	15
3.4 Predicting the Class	16
3.5 Consensus Methods	16
3.6 Interactive Visualization	18
Chapter 4 Experiments and Results	22
4.1 Data Collection of the Research Articles (Topic and Abstract Information)	22

4.2	Evaluation Measures	23
4.3	User Study	24
4.3.1	Population	24
4.3.2	User Tasks — Machine Learning System	25
4.3.3	User Tasks — Interactive Visualization System	25
4.3.4	Results — Machine Learning System	25
4.3.5	Results — Interactive Visualization System	26
Chapter 5	Conclusion	29
5.1	Future Work	29
Bibliography	32
Appendix A	Implementation Details	35
A.1	Data	35
A.1.1	Creating Microsoft Academic API Key to use Extractor tool	36
A.1.2	List of topics and keywords	37
A.1.3	Data Size	37
A.2	NSERC Research Topics and Keywords	39
Appendix B	User Study Details	47
B.1	Introduction	47
B.1.1	Study Population and Plan	47
B.1.2	Research Question	48
B.1.3	User Recruitment	48
B.1.4	Informed Consent Process	49
B.1.5	Study Design	49
B.1.6	Data Analysis	50

List of Tables

4.1	No. of Correct Topics Predicted by the Machine Learning System as evaluated by the Users. Here second column refers to the number of correct topics that each user thinks are correct (chosen out of 3, which is the number of predicted topics from the machine learning system). std: Standard Deviation.	26
4.2	Time spent (in seconds) by users in answering questions using the Visualization system and Database/Spreadsheet. std: Standard Deviation.	27
4.3	Post-Condition Questionnaire (Completed by each user at the end of their session)	28
A.1	Statistics of Data	35
A.2	NSERC Research Topics and the corresponding Keywords for the Computer Science Evaluation Group	46
B.1	Visualization System Test Quizzes (for making users interact with the visualization System and Spreadsheets)	50

List of Figures

1.1	ACMDSP Speaker Data	2
1.2	ACMDSP Lecture Data (Lectures Given)	3
1.3	Data Extraction using the Microsoft API	4
1.4	Topic(s) Prediction Architecture	5
1.5	Interactive Visualization Architecture (Querying and Response)	6
3.1	Methodology for classifying document into research topic(s) [22].	12
3.2	Sample input of text for Wikification to Wikifier	14
3.3	Wikified text from Wikifier	15
3.4	Categories for a given Concept	16
3.5	Methodology for classifying with the Class with Max. Probability as the Consensus Method. Each classifier $\mathbb{C}\mathbb{L}_x$ generates a list of (class, probability) pairs $(C_i, P_x(C_i))$. The output class $C^{mp} = \max_{x,i} P_x(C_i)$, where $i = 1 \dots C $ and $x \in \{bow, boc, bok\}$.	17
3.6	Methodology for classifying with the Class with Max. Vote as the Consensus Method. Each classifier $\mathbb{C}\mathbb{L}_x$ returns a single class C_x . Voting picks the majority class among the C_x over $x \in \{bow, boc, bok\}$. If the C_x are all different from one another, voting picks one of them in random.	18
3.7	Linear Regression as the Consensus Method. Each classifier $\mathbb{C}\mathbb{L}_x$ generates a list of (class, probability) pairs $(C_i, P_x(C_i))$, where $i = 1 \dots C $ and $x \in \{bow, boc, bok\}$. We thus have three vectors of probabilities $[[P_{bow}(C_i)], [P_{boc}(C_i)], [P_{bok}(C_i)]]$. This is the input to the linear regressor, while the output is an one-hot vector of dimensionality $ C $, where the 1 corresponds to the true class. The linear regressor is trained on the training data we have available. Given a test document, we put it through the three base classifiers, compute the probability vectors from each classifier, then we form the linear combination of these vectors, and return the class corresponding to the maximum element of the output of the regressor.	19
3.8	Interactive Visualization of the ACMDSP database	19

3.9	Visualization of Details About Speaker/Topic/Lecture of the ACMDSP Database	20
3.10	Visualization of Query And Response by the ACMDSP Database (Speakers for a Particular Topic are Returned)	20
A.1	Microsoft Academic Extractor Tool GUI	37
A.2	List of ACMDSP topics and corresponding NSERC keywords .	38
A.3	List of speakers and their details	38
A.4	No. of Lectures on Offer	39
A.5	No. of Lectures Given	39
A.6	Training Data — Title, Abstract and corresponding ACMDSP Topic	40

Abstract

Association for Computing Machinery (ACM) is an international learned society for computing. ACM operates the Distinguished Speaker Program (ACMDSP). ACMDSP maintains a list of speakers, who can be invited to deliver lectures on Computer Science topics at different locations worldwide. Currently, speakers' lectures are classified into topics manually and ACMDSP committee accesses the speaker and lecture data directly through the database. This thesis is attempting to make it more intuitive to access the database through a visualization system, and in classifying the lectures on offer into topics. It uses Google Map to visualize the speaker, topic and lecture data. It displays the speaker's location and contact details on Google Map.

Each lecture delivered by the speakers is assigned to one or more topics from the set of topics defined by the ACMDSP committee. The problem of categorizing lectures into topics is similar to the problem of categorizing research papers into topics. Hence, for each topic, we have manually associated a set of keywords from the NSERC list of research topics. These keywords are used to create training sets for each topic. Title and abstract information of these research papers along with a lecture topic are used to train the machine learning models, which classify each lecture title and abstract into one or more topics of a predefined topic structure.

This thesis uses three document representations, based on bag of words, bag of concepts and bag of categories. We have used three consensus methods, which include linear regression, class with maximum probability and voting based. Each of these methods is a consensus method in itself and every individual consensus method forms an agreement to predict a topic.

This thesis expanded on the previous classification model based on semantic representations of lecture titles/abstracts that can classify a large set of lectures into topics. Previous work used the topics to construct the training data. However, this thesis used the NSERC keywords to describe the ACMDSP topics and construct the training data. The classifier can predict up to three topics for a single Lecture.

List of Abbreviations Used

ACM	Association for Computing Machinery
ACMDSP	ACM Distinguished Speaker Program
API	Application Programming Interface
BOW	Bag of Words
DOI	Digital Object Identifier
Fig.	Figure
GSC	Grant Selection Committee
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ID	Identity
IDF	Inverse Document Frequency
LDA	Latent Dirichlet Allocation
NSERC	Natural Sciences and Engineering Research Council
PI	Principal Investigator
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TF	Term Frequency
TFIDF	Term Frequency — Inverse Document Frequency
UI	User Interface
URL	Uniform Resource Locator
WoS	Web of Science

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr. Evangelos Milios and co-supervisor Dr. Fernando Paulovich, for their constant guidance and encouragement. It was a great opportunity to work under your supervision. Thanks for trusting me and giving an opportunity to learn and apply my skills. Many thanks for accepting me for research with you. I would like to thank lab members for their support, timely guidance, for extending help whenever needed during the research.

I would like to sincerely thank Dr. Vlado Keselj and Dr. Luis Torgo for their critical comments that helped improve the quality of the thesis. I would also like to thank my family, friends, and colleagues at the Faculty of Computer Science, Dalhousie University for their encouragement and moral support which has helped me throughout the journey of this program.

Chapter 1

Introduction

There are many research papers published every year. These papers need to be classified into categories/topics. Traditionally, these papers are read and are manually classified. These topics are defined by various committees and organizations. They change as the field progresses with time. However, this process is time-consuming and can be erroneous as well. Recently, there has been an increase in the number of research papers and classifying these papers into the correct class of the research category helps in creating a group of papers which belong to certain categories. Hence, there is a need to classify these papers into topics automatically.

One of our possible user is the ACM DSP ¹ committee. ACM is the Association for Computing Machinery. ACM DSP is the ACM Distinguished Speaker Program, which has a set of speakers which offer lectures on various topics. These lectures are given on demand by various organizations, companies, and educational institutions. Speakers have to travel depending on their resident location and the location of the event. Speakers are invited to deliver lectures on Computer Science topics at various locations worldwide. Committee manages speaker recruitment based on supply and demand of speakers.

ACM DSP committee has a set of pre-defined topics and speakers need to classify their delivered lectures into those topics. Currently, speakers' lectures are classified into topics manually. The thesis is trying to semi-automate the classification by creating a machine learning system, which will classify lectures into topics and these topics are provided to the speaker as recommendations. Speakers can add or remove any topics from the set of the recommended topics.

Moreover, we have the speaker data, which is a list of speakers with associated lecture titles and abstracts, classified by topics as shown in Fig. 1.1. Lecture data contains list of lectures given, their location, lecture date, speaker delivering the

¹<https://speakers.acm.org/>

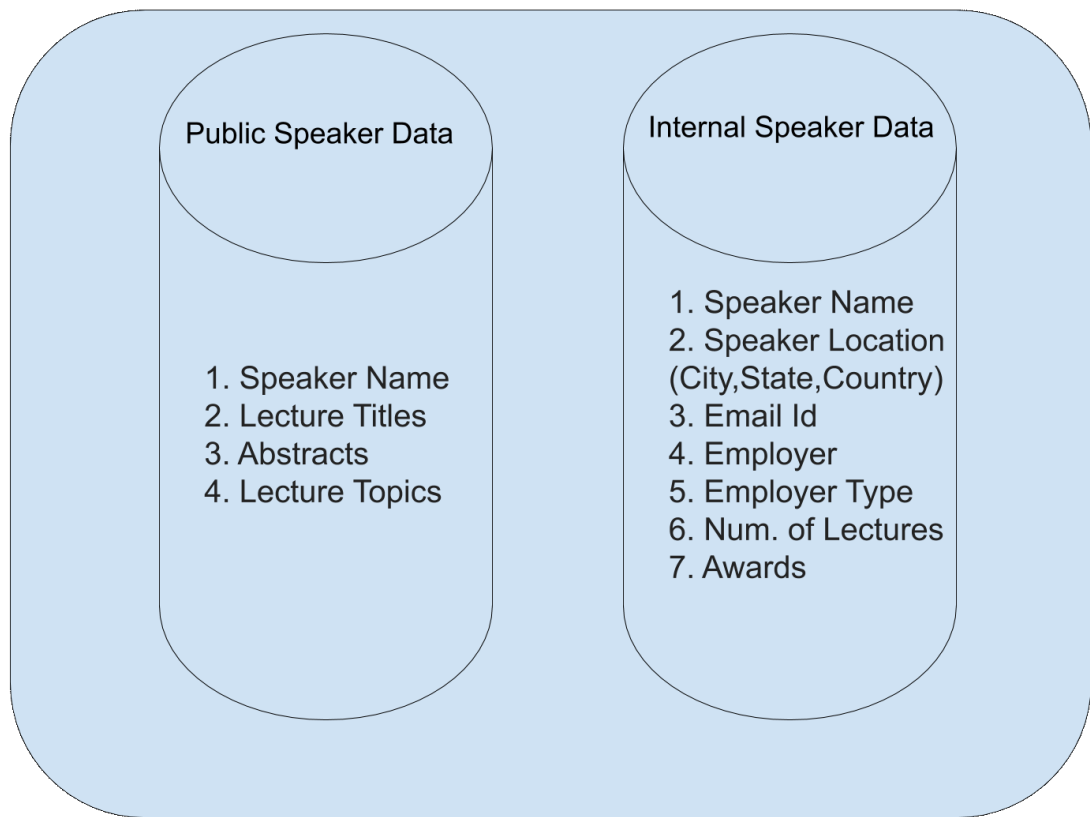


Figure 1.1: ACMDSP Speaker Data



Figure 1.2: ACMDSP Lecture Data (Lectures Given)

lecture, and the lecture title/abstract as shown in Fig. 1.2. Currently, data is being accessed directly through the database by the ACM DSP committee. We are attempting to build an interactive visualization system that facilitates the interaction of the ACM DSP committee with the database.

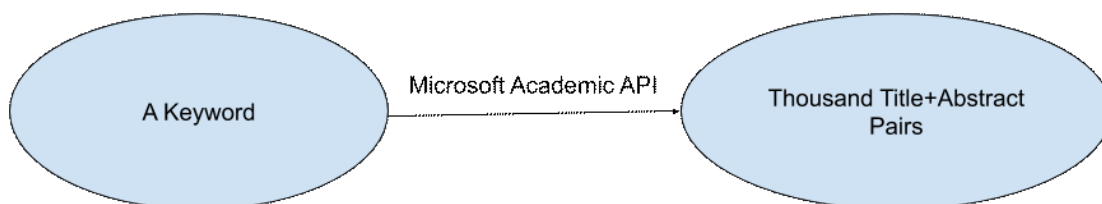


Figure 1.3: Data Extraction using the Microsoft API

Title and abstract of a lecture delivered by the ACM DSP speaker is similar to the title and abstract of a research paper. Moreover, the problem of categorizing lectures delivered by various speakers into topics is similar to the problem of categorizing research papers into topics/categories. Hence, we need to collect a list of titles and abstracts of research papers, which will be the training data for creating a machine learning system. For each topic, we have manually defined a set of keywords from the NSERC list of research topics. These keywords are used to find the research papers for each topic. We have used the Microsoft Academic API for this purpose. A list of keywords is passed to the Microsoft Academic API and it returns the list of titles and abstracts from the research papers corresponding to each keyword.

We have applied a filter to collect only up to a thousand research papers for each keyword. A number less than thousand would lead to less training data and there was a need to keep an upper limit on the number, hence the number thousand was chosen. We have also applied a filter to collect research papers only for the last five years. The extracted data contains a topic, list of title and abstract of the research paper corresponding to that topic. The data extraction architecture using the Microsoft API is shown in Fig. 1.3.

We further used this data to train our machine learning model. In the machine learning model, we use title and abstract from the research paper as our input and the research topic as our output. We have used various ways to convert title and abstract into document representations. Mainly three document representations are

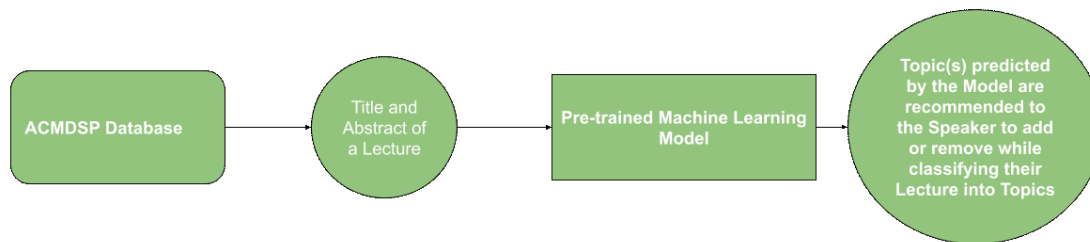


Figure 1.4: Topic(s) Prediction Architecture

used, which includes bag of words, bag of categories, bag of concepts. In the bag of words model, title and abstract is converted into a list of words which act as features. The bag of concepts model is used to overcome the drawbacks of the bag of words model. In the bag of concepts model, Wikification is used. Wikification includes disambiguation of the terms using the Wikipedia knowledge base. Wikification refers a term to the correct Wikipedia article. These terms are known as concepts. Bag of concepts is an enriched representation over bag of words. We have used the Wikipedia Miner Toolkit, Wikifier by University of Illinois [24, 4] to extract concepts for a given document. A paper on Biomedical literature classification [8] uses bag of concepts and mentions that most of the classification approaches follow the traditional bag of words approach, which has many limitations. Bag of words approach suffers from synonymy and polysemy. Moreover, their weights are just based on their frequency of occurrence. Hence bag of concepts is a much better approach than bag of words for this scenario.

The third document representation is bag of categories. The concepts identified in the bag of concepts model have some amount of noise during the classification. Hence there is a need for this third representation. Wikipedia is very densely connected network of information. Each Wikipedia article is connected with various categories and other articles. Hence the categories mentioned in the Wikipedia article represent the breadth and the articles mentioned in the Wikipedia article represent the depth. We have used the Wikipedia based tool, Sunflower [19], to extract categories for concepts.

For the purpose of vectorization, we have used *TFIDF*, which refers to the term frequency inverse document frequency. Further, we have used the consensus methods. Three consensus methods are used, which include linear regression, class with

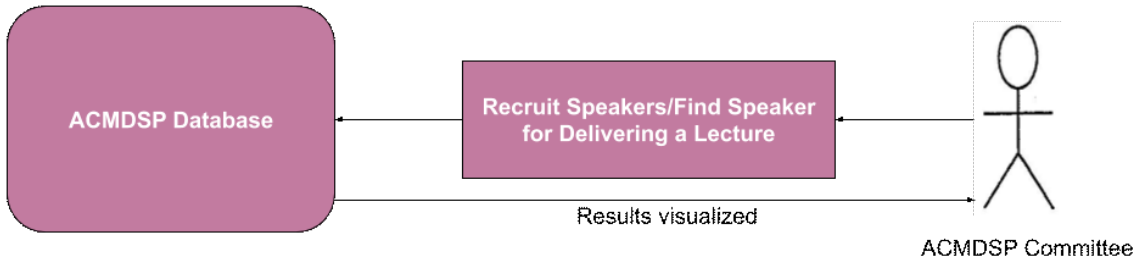


Figure 1.5: Interactive Visualization Architecture (Querying and Response)

maximum probability, voting based. The topic prediction architecture is shown in Fig. 1.4.

Our current methodology is using bag of words, bag of concepts and bag of categories in the approach. It combines result from three classifier together. The reason for not using standard ensemble methods or just one classifier is because the combination of classifiers is better than any individual classifier, as demonstrated in the previous research work in [22].

For the visualization system, we have used the Google Map API to visualize the speaker and lecture data. It displays the speaker's location and contact details on the Google Map. Further, it provides search options and filters to access the required information from the database. It also helps ACMDSP Committee in recruitment of new speakers based on their supply and demand. The interactive visualization architecture is as shown in Fig. 1.5. This approach is more intuitive and faster for a user than the traditional search in a database.

Chapter 2

Related Work

2.1 Introduction

There has been a lot of research work on finding categories/topics for some expertise. In this chapter, we will discuss the previous research work being done in this area.

The methodology that was used to collect the data related to title and abstract of research papers and the methodology followed to create the machine learning model was adapted from [22]. Moreover, there are many other research papers that discuss interesting methodologies and results. Klout topics for Modeling Interests and Expertise of Users Across Social Networks [6] mentions that it uses Klout Topics, which is a lightweight curated ontology and is designed to model text and user topics across the social networks. It has Topic Nodes, Topic Edges, and Topic Display Names. Each Topic Node has three components — `topic_id`, `slug`, and `metadata`. `Topic_id` is a unique numerical identifier for the topic. `Slug` is a human-readable English label, usable in a URL. `Metadata` includes both the Wikidata entity ID and Freebase `machine_id` of the closest corresponding entity. Topic Edges are structured as a directed graph with all branches connected to each other. Each edge contains three components — `edge_id`, `source_topic_id`, and `destination_topic_id`. `Edge_id` is a unique numerical identifier for the edge. `Source_topic_id` is the `topic_id` of the parent topic and the `destination_topic_id` is the `topic_id` of the child topic. A Topic Display Name has four components — `topic_id`, `language`, `display_type`, and `display_name`. `Topic_id` is the unique identifier for the topic. The term `language` here refers to the language of the given `display_name`. `Display_type` is a flag to indicate if the topic should be displayed in the given language and `display_name` is the human-readable form of the topic name.

User Modeling of Skills and Expertise from Resumes paper [18] mentions the REMA algorithm. REMA (Resume Expertise Modeling Algorithm) is an extension of the user modeling algorithm RAMA (Reinforcement and Aging Modeling Algorithm).

REMA takes data from a resume document as input and produces an expertise model. The expertise model details the expertise topics along with a weight indicating the level of competency. The main two key insights for this algorithm are — first, expertise is the cumulative result of the various learning events and second, one’s skills and knowledge can become outdated with time if not reinforced by learning.

Trends in news content in large digital news archives are analyzed using the Latent Dirichlet Allocation (LDA) topic modeling [15]. Similarly, latent dirichlet allocation for topic modeling [23] can be used for analyzing trends on Twitter. YouTube comments can be used for the detection of textual cyberbullying [5]. It uses the *TFIDF* for vectorization and shows that the label-specific classifiers are more effective than multiclass classifiers for this particular scenario.

Aspect-based sentiment analysis [3] of Arabic hotels reviews, uses both Deep Recurrent neural network and support vector machine for this purpose. It concludes that SVM approach outperforms the deep RNN approach for this particular scenario.

2.2 Data Collection

There is lot of work being done in this area using different tools/APIs. For example, the StackExchange API is being used to collect the data related to forum posts from stackoverflow website in the topic facet modeling and semantic visual analytics for online discussion forums paper [12]. Similar to post extraction, there are many other papers which mention different ways to extract the research articles [28] e.g. Microsoft Academic API, Google Scholar, ArnetMiner, DBLP, Citeulike, and CiteSeer. This paper also displays a visual analytics prototype displaying different filters for speakers, topics and time range, which is considered more user-friendly for interacting with the system.

Microsoft Academic can be used for bibliometric analyses [14]. This paper mentions and compares the meta data being collected from Microsoft Academic and Google Scholar. Comparison of publication and citation coverage of various resources [11] such as Microsoft Academic, Google Scholar, Scopus and the Web of Science emphasizes importance of the Microsoft Academic. Hence, we decided to use Microsoft Academic API in the thesis. Microsoft Academic finds 90% of the articles with DOI and 89% without DOI as mentioned in the paper on the search capability

of Microsoft Academic [26]. It also finds that the remaining articles are either not indexed by Microsoft Academic or indexed with a different language version of their title.

In another paper, the coverage of Microsoft Academic is measured by analyzing the publication output of a university [13]. The coverage of Microsoft Academic was assessed and compared with two benchmark databases, Scopus and Web of Science (WoS). Citation counts were analyzed, and issues related to data retrieval and data quality were examined.

2.3 Disambiguation to Wikipedia

Wikipedia is a source of inter-linked knowledge. Wikipedia hyperlink graph for relatedness and disambiguation is studied in the paper [2]. It shows that Wikipedia has categories which represent breadth and the categories are linked to other categories, which represents the depth. Wikification is the task of identifying the Wikipedia articles and categories in the given text terms. Wikipedia-based concepts can improve text classification results over BOW as shown in the paper [7]. These results show improved performance over bag of words representation using Wikipedia enriched representation. There are many tools that can do Wikification. One such tool is an open-source toolkit for mining Wikipedia by Milne, D. and Witten, I. H [21]. But this tool has some limitations. It does use the Wikipedia articles and the link structure but fails to use the underlined contextual knowledge. There is one more tool which is created by the University of Illinois[4]. This new Wikifier has been optimized to outperform the state of the art system. This new global system is known as Glow.

2.4 Visualization

There are many visualization techniques possible to visualize the type of data that we have. Some of these visualizations include heatmaps, the Google Map and Three Dimensional Extrusions. Spatial visualization of location-based data using the ggplot2 in R [16], works by representing the latitude and longitude information of a particular location in the maps. It further uses the Google Map API to represent this

data. Geospatial data of the conference presenters are visualized in a paper that discusses the geospatial visualization using the Google Map [29]. Since our data is also location-based, we found that the best way to visualize is to represent the location data on the Google Map.

There are many more papers that discuss using the Google Map API for visualization of geographical data. An article reports the experience of using the Google Map API and the Google Chart Tools in a data visualization course at Georgia State University [30]. It shows that students found these tools and APIs to be very simple to use. Hence, the Google Map API has been used in the thesis for visualization of geographical data of various ACM DSP speakers. Geospatial data can be visualized with the Google Map or the Google Earth as suggested in the paper [25]. This paper discusses the two of the most widely adapted formats for visualizing data in the Google Earth. These two formats are — Geography Markup Language (GML) and Google's Keyhole Markup Language (KML). It also mentions a three-step approach for visualizing and publishing geospatial data. The first step is to model geospatial data in GML. This step is followed by converting GML data into KML and the last step is to publish ML data on the Web and visualize it with the Google Earth.

Chapter 3

Methodology

3.1 Introduction

The methodology that was used to collect the data related to title and abstract of research papers and the methodology followed to create the machine learning model, adapted from [22], is described in this Chapter. We have created one classifier for N classes. Here classes refer to the list of topics and the value of N is forty-two. This value of N comes from the number of ACM DSP topics. We have used the keywords from the NSERC research topics [10], treated as an intermediate representation, to describe the higher-level topics defined by the ACM DSP committee. The reason to use the NSERC keywords to query Microsoft Academic instead of using the ACM DSP topics directly is because the NSERC keywords are large in numbers and this helps in extracting very broad set of data for training the machine learning model. Moreover, our research is only focusing on the NSERC keywords and the NSERC research topics are not being used. A detailed description of NSERC, their evaluation groups, research topics and keywords is mentioned in the Table A.2.

These keywords are used as input to the Microsoft Academic API to collect title and abstract of the research papers, to be used as training data. The list of title and abstract extracted for each keyword corresponding to a topic are combined together to create the training data for that topic. This complete data extracted using Microsoft Academic API is divided into 70%/30%, where 70% is the training data and 30% is the testing data. Title and abstract together are considered as a document. During data extraction, there is a possibility of extraction of same title and abstract for two different topics. However, since we have up to three predicted topics, so if the machine learning system's output matches with any of these topics, it is counted as a match and it contributes towards accuracy of the machine learning model. There are three document representations that we have used namely, bag of words, bag of concepts and bag of categories model. Out of these three representations, the last two

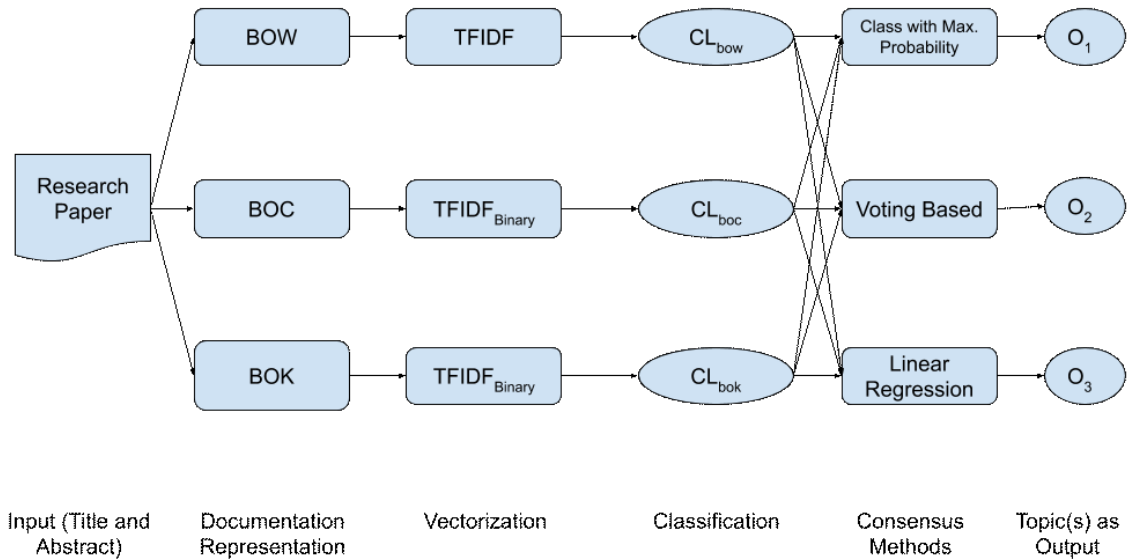


Figure 3.1: Methodology for classifying document into research topic(s) [22].

representations, bag of concepts and bag of categories, are based on the Wikipedia concepts and categories respectively. Further, we have used *TFIDF* to convert document representations into vectors. The output from *TFIDF*, which is a vector, is passed over to the Support Vector Machine classifier. Further, three consensus methods are used, which are: linear regression, class with maximum probability, voting based. Consensus method combines the output from three classifiers to predict the research topic(s) for a title and abstract. The machine learning system can predict up to three topics. The complete methodology for classification of the document into the research topic(s) is shown in Fig. 3.1.

For the interactive visualization part, we have converted the data in the database into a visual form. Since we have the option to search for a particular speaker by name or topic hence we have provided a filter search speaker by their name or their expertise/topic or year. For the visualization, we have used the Google Map API. We have converted a piece of area information into latitude and longitude using the Google Map API. We have created an HTML page that displays the Google Map on the left side and all filters and search options on the right side.

3.2 Document Representation

Since the data that we have is in textual form, we need to convert this data into numbers. Generally, the text data is represented as bag of words, which is followed by vectorization using *TFIDF*. TF stands for term frequency and IDF stands for inverse document frequency. But since bag of words has many drawbacks, we have used bag of concepts as well as bag of categories. Both, bag of concepts and bag of categories, use Wikipedia knowledge.

3.2.1 Bag of Words

Bag of words is a traditional approach to convert text into words for text analysis. This helps in converting text into features. Before conversion of text into features, preprocessing and cleaning of data is done. Preprocessing includes removing English stopwords, stemming and removing duplicate words. Further, vectorization is done using *TFIDF*. $TF_{t,d}$ stands for term frequency, which is the number of occurrences of Term t in document d . Document here refers to the combination of title and abstract. There can be some terms which may have high frequency and may dominate during classification. Modified weighing scheme, called log normalization ($1 + TF_{t,d}$), is used to diminish this effect. *IDF* stands for inverse document frequency, which is calculated to know if a word is rare or common. Finally, the product of *TF* and *IDF* is calculated for each term belonging to the document. D is the set of documents, that is, the combination of title and abstract corresponding to topics. This combination of title and abstract is extracted using the Microsoft Academic API.

$$IDF_{t,D} = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \quad (3.1)$$

$$TF.IDF_{t,d,D} = TF_{t,d} * IDF_{t,D} \quad (3.2)$$

The bag of words representation is a simple approach but has many limitations. One such limitation is that it is further away from the meaning of the document than other representations, i.e. bag of concepts and bag of categories.

3.2.2 Bag of Concepts

The bag of words approach has many limitations hence we follow another approach known as bag of concepts. This acts as an enriched bag of words approach. Bag of concepts approach uses Wikipedia knowledge. This approach follows the process of Wikification. Wikification removes disambiguation of the terms using Wikipedia knowledge base. It refers a term to the correct article using Wikification. These terms are known as concepts.

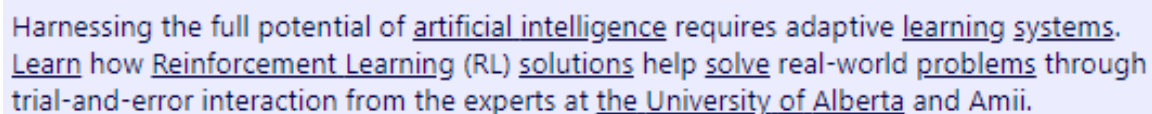
We have used Wikipedia Miner Toolkit, Wikifier, to extract concepts for a given document. New vector representation is created for each document using *TFIDF*. The set of features also includes words that are not wikifiable i.e. the words which are not Wikified. For each document, a Boolean term frequency is calculated in this representation. Only a single instance is retrieved for every occurrence of the concept. Each concept has a name which is linked to the Wikipedia article and has a unique identification number.

Each identified concept is assigned a score based on the similarity to the text. This score signifies the probability of the text being similar to the concept. Although Wikipedia is considered a good source for the purpose of disambiguation, there can be concepts which are not relevant to the text. The list of concepts after Wikification needs to be pruned by selecting appropriate probability threshold. Wikification using the Wikipedia miner toolkit disambiguates into Wikipedia concepts. If we have more than one term in the concept then those are combined using underscore “_” symbol.

Further, concepts in each document are represented as vectors using *TFIDF* vectorization. The only difference between *TFIDF* vectorization in bag of concepts and bag of words is that in bag of concepts, each time is assigned a score of either zero or one based on if the term exists or not. A sample input of text and Wikified text is shown in below Fig. 3.2 and Fig. 3.3.

Harnessing the full potential of artificial intelligence requires adaptive learning systems. Learn how Reinforcement Learning (RL) solutions help solve real-world problems through trial-and-error interaction from the experts at the University of Alberta and Amii.

Figure 3.2: Sample input of text for Wikification to Wikifier



Harnessing the full potential of artificial intelligence requires adaptive learning systems.
 Learn how Reinforcement Learning (RL) solutions help solve real-world problems through
 trial-and-error interaction from the experts at the University of Alberta and Amii.

Figure 3.3: Wikified text from Wikifier

3.2.3 Bag of Categories

Wikipedia is an online encyclopedia manually built over the years. It is a densely linked network of information and is a multilingual online encyclopedia. Wikipedia is based on open collaboration through a wiki-based content editing system, which makes it a good choice to use for the thesis. Each Wikipedia article is not only related to a Wikipedia concept, but also categorizes itself. Wikification is the process of converting terms in text into concepts. However, there is some noise during Wikification because identification of concepts by Wikipedia is not completely correct. Hence there is a need to enrich the bag of words representation further.

Every Wikipedia article is connected with many categories and other articles. Categories mentioned in the Wikipedia article represent the breadth and the categories are linked to other categories, which represents the depth. This tool extracts categories for corresponding concepts. Based on the similarity of the concept and the categories, a score is assigned to each category. If the name of the category has more than one term then the terms are combined using underscore “_” symbol.

As an example, the concept “pattern recognition” has categories “machine learning”, “sciences” and many more as shown in Fig. 3.4. The breadth and depth of the categories can be managed by maintaining a threshold. Maintaining a threshold of the score is important, since there can be scenarios where irrelevant categories are extracted. Further, the bag of categories representation is vectorized using *TFIDF* as mentioned in the other two document representations, bag of words and bag of concepts.

3.3 Classification

As discussed above, there are three documentation representations — bag of words, bag of concepts and bag of categories. Further, *TFIDF* vectorization is used and this

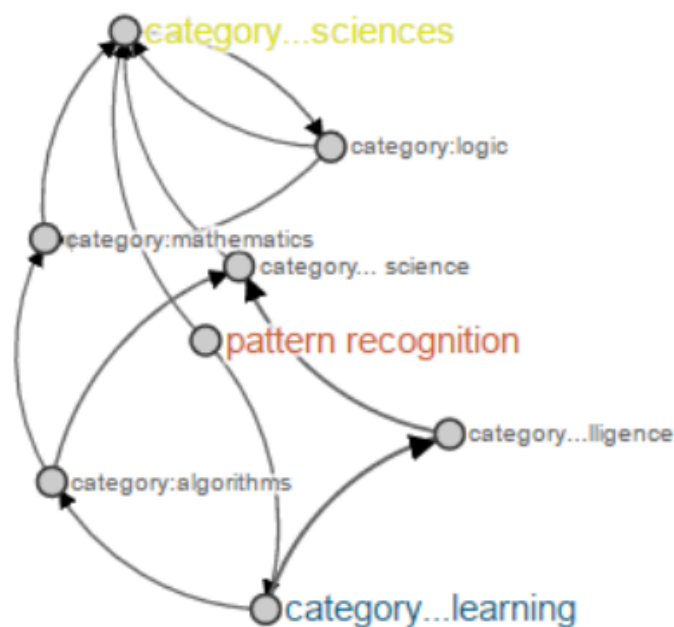


Figure 3.4: Categories for a given Concept

is used as an input to the classifier for training. The corresponding topics are passed as an output to the classifier. Support Vector Machine¹ is used as a classifier. A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. It outputs an optimal hyperplane which categorizes given labeled training data. It outputs the predicted topics/classes² and the probability of each topic.

3.4 Predicting the Class

The output from the classifier are the topic(s) and the probabilities of the topics. This output is passed over to three different consensus methods; i.e., class with the maximum probability, class with maximum votes, and by training a linear regression. This consensus method is used for agreement on the topics.

3.5 Consensus Methods

So far we have converted text into various document representations using bag of words, bag of concepts and bag of categories. These representations have converted

¹<https://scikit-learn.org/stable/modules/svm.html#multi-class-classification>

²Parameters used are: gamma='scale', C=1.0, kernel='rbf', probability=True

text into features. Further, we have used *TFIDF* vectorizer to represent document representations as vectors. Then we have used a classifier for each representation. Now, there are three predicted classes from these three classifiers. The output from each of these classifiers is combined using different approaches based on some agreement. Consensus method is the way to combine individual classifiers.

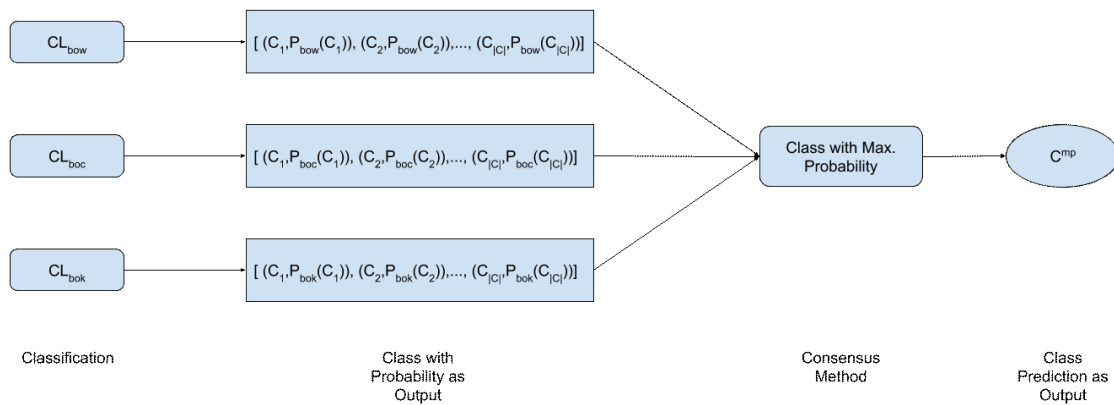


Figure 3.5: Methodology for classifying with the Class with Max. Probability as the Consensus Method. Each classifier $\mathbb{C}L_x$ generates a list of (class, probability) pairs $(C_i, P_x(C_i))$. The output class $C^{mp} = \max_{x,i} P_x(C_i)$, where $i = 1 \dots |C|$ and $x \in \{bow, boc, bok\}$.

The first method outputs a class with maximum probability by combining the predicted classes from different classifiers for different document representations. It is shown in Fig. 3.5. The second method combines the predicted class from different classifiers to output a class with maximum votes. Here, votes refer to the number of times the class is being predicted by various classifiers. However, if there is any tie in the number of votes, then either of the classes in the tie is returned. It is shown in Fig. 3.6. In the third approach, linear regressor³ is trained on a training set that consists of probability vectors from each classifier for each document in our training set and the target value is the one-hot vector representation of the class label. All

³https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

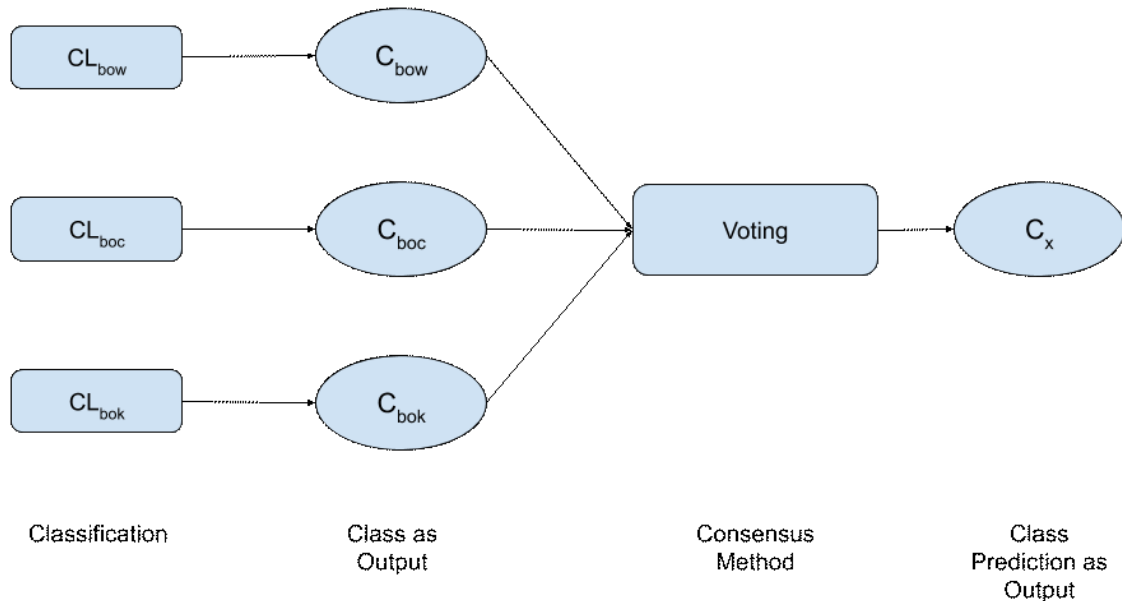


Figure 3.6: Methodology for classifying with the Class with Max. Vote as the Consensus Method. Each classifier CL_x returns a single class C_x . Voting picks the majority class among the C_x over $x \in \{bow, boc, bok\}$. If the C_x are all different from one another, voting picks one of them in random.

probability values are continuous and the target values are made ordinal using one-hot encoding⁴. Each class is converted to a sequence of bits of length equal to the number of classes. Only one bit at a specific position is “On” and all others are “off”. The target is represented as forty-two columns, which is the number of ACMDSP topics. A On bit at a specific position corresponds to the label of a ACMDSP topic. The methodology is shown in Fig. 3.7.

3.6 Interactive Visualization

ACMDSP has a set of speakers which deliver lectures at various locations throughout the world. Searching for a speaker for a particular lecture on a topic requires a lot of effort as currently this is being done manually by searching in the database.

This thesis has converted the data in the database into a visual form as in Fig. 3.8. The speaker and topics data is visualized on the map. For the purpose of visualization, we have used the Google Map API. Various speakers are visualized using their location

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>

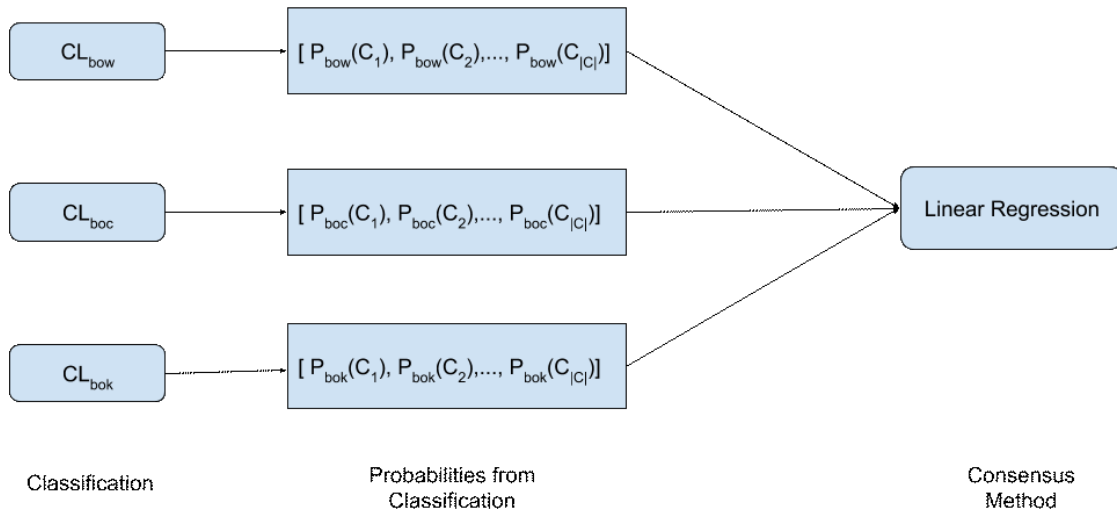


Figure 3.7: Linear Regression as the Consensus Method. Each classifier CL_x generates a list of (class, probability) pairs $(C_i, P_x(C_i))$, where $i = 1..|C|$ and $x \in \{bow, boc, bok\}$. We thus have three vectors of probabilities $[[P_{bow}(C_i)], [P_{boc}(C_i)], [P_{bok}(C_i)]]$. This is the input to the linear regressor, while the output is an one-hot vector of dimensionality $|C|$, where the 1 corresponds to the true class. The linear regressor is trained on the training data we have available. Given a test document, we put it through the three base classifiers, compute the probability vectors from each classifier, then we form the linear combination of these vectors, and return the class corresponding to the maximum element of the output of the regressor.

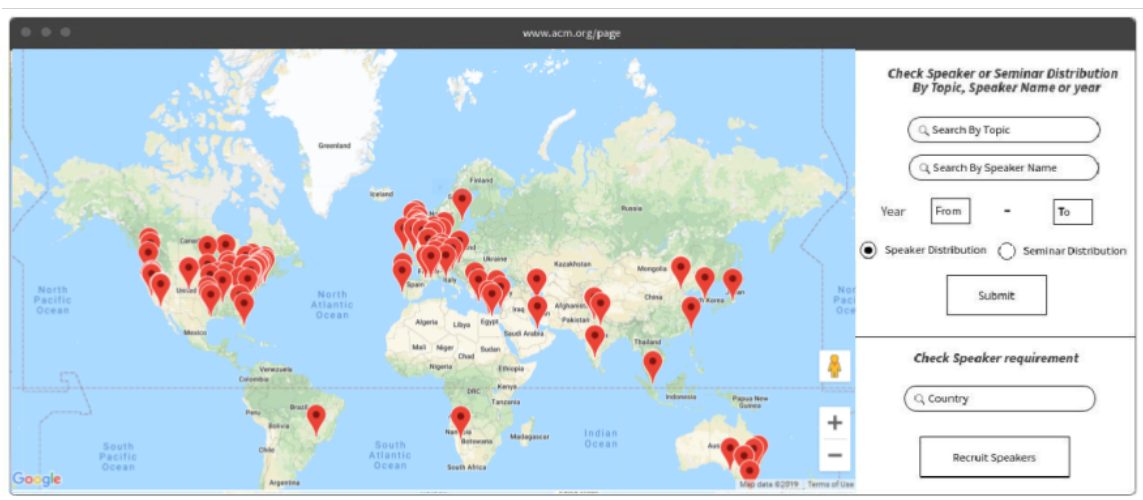


Figure 3.8: Interactive Visualization of the ACMDSP database

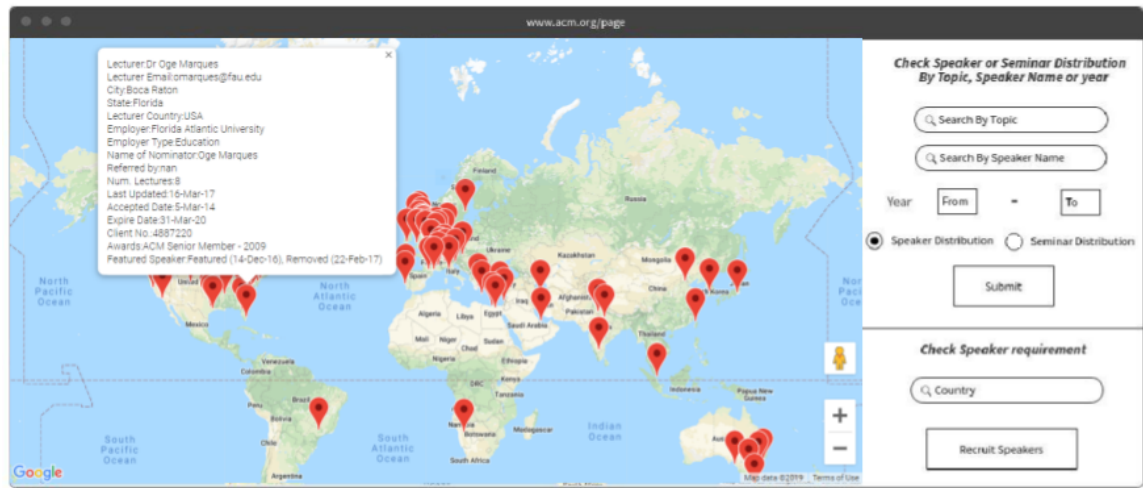


Figure 3.9: Visualization of Details About Speaker/Topic/Lecture of the ACMDSP Database

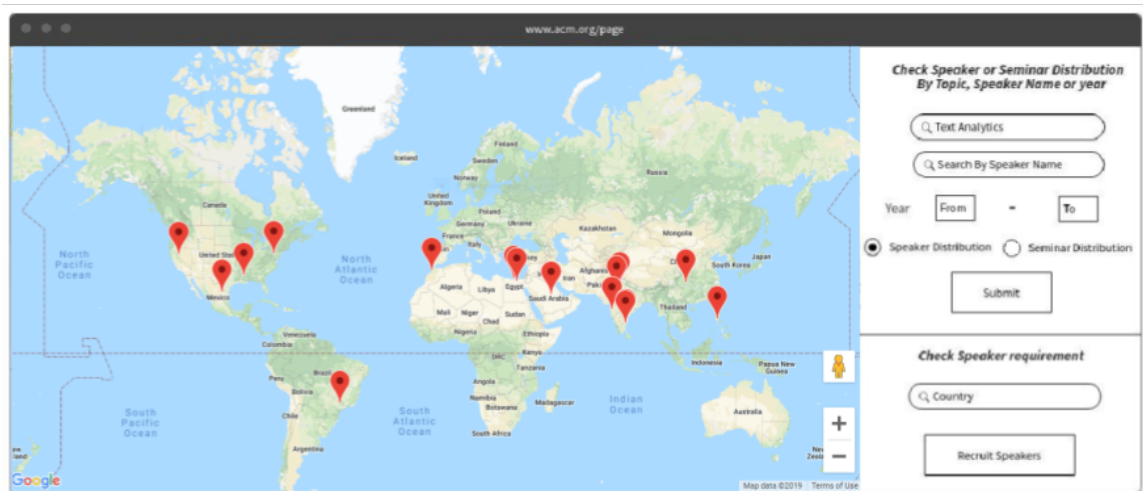


Figure 3.10: Visualization of Query And Response by the ACMDSP Database (Speakers for a Particular Topic are Returned)

data on the Google Map and we have provided users with the option to search a speaker based on their name or their lecture topic. This interactive visualization of the speakers and topics is shown in Fig. 3.8. All locations are clickable and provide more detail when clicked as shown in Fig. 3.9.

The Google Map API along with the search options or filters on the right side are displayed within an HTML page. Users can use this visualization to search for any speaker by name or topic as shown in Fig. 3.10. This data is searched ⁵ in the database and the results are returned ⁶. The returned results are a list of locations which are visualized on the Google map. Our user study has shown that the search through the Google Map API is much faster than searching directly within the database/spreadsheet inspection.

⁵This HTML page is connected with the back end database using the Flask framework <http://flask.pocoo.org/>.

⁶This data entered by the user inside the text boxes in the form present on the right side of the web page is passed to the back end Python script using the Flask framework.

Chapter 4

Experiments and Results

This chapter discusses how the training data was collected, the various experiments performed, user study and the results obtained after the study.

4.1 Data Collection of the Research Articles (Topic and Abstract Information)

Our aim is to find the research articles corresponding to the topics in our database. For the thesis, we have considered the title and abstract of the research articles extracted based on the keywords for each topic. Title and abstract information is used to build the training data for the machine learning models. The reason to use title and abstract information for the thesis is that because an abstract of the research articles describes the research in a precise and summarized way.

There are various ways to search and extract the research Articles corresponding to the keywords e.g. Microsoft Academic API, Google Scholar, ArnetMiner, DBLP, Citeulike, and CiteSeer. Out of all these options we have used the Microsoft Academic API.

To use the Microsoft Academic API, a key is required ¹. An application with a graphical user interface was designed to extract research articles based on the search keyword, the threshold on the number of research paper information extracted, the date range (the starting date and the ending date) in the text boxes available in the graphical user interface. It also allows a user to choose the attributes from — title, abstract, author name, date. ²

Since there are a large number of research articles present online, we need to restrict our search and extraction capabilities. Hence, we have applied a threshold of

¹The Python script, written to extract title and abstract information of research articles based on the keywords, was connected with a graphical user interface using the Tkinter library.

²The GUI provides a “Collect Data” button, which begins the data extraction using the Microsoft Academic API.

a thousand research articles corresponding to each keyword. Apart from applying restriction on the number of research articles extracted for each keyword, we have also applied a restriction on the publication year of the research article. We consider the research articles which are published within the last five years only. The most challenging part of the data extraction step is to extract the complete abstract and title information. The Microsoft Academic API has been used in such a way that it avoids title and abstract with partial information. Hence all the data, title, and abstract information collected using this API is complete. Title and abstract information extracted corresponding to each keyword is combined together. The combined information of the title and abstract of a research article acts as an input for training the machine learning model and the topic name/lecture topic acts as an output.

Title and abstract information is preprocessed and cleaned to remove any English stopwords. After removing the stopwords, stemming is done which is followed by removing duplicate words. Later, various document representations are applied to the text present in title and abstract information. These document representations are bag of words, bag of concepts and bag of categories. Stemming is done for all the words except the concepts. Stemming is not done in the case of bag of categories. Further, vectorization is applied to these document representations. We have used *TFIDF* for the conversion of features into vectors.

4.2 Evaluation Measures

We have used various evaluation measures such as Precision, Recall, and F-measure for measuring the performance of the machine learning model on the testing data. The accuracy of the model is 76.42%. A prediction is counted as true, if any of the three predicted topics match with the actual topic.

The number of true positives over the sum of the number of true positives and the number of false positives is known as precision. Precision is a good measure to determine if the cost of false positive is high. The precision is 75.48%.

$$P = \frac{|T_p|}{|T_p| + |F_p|} \quad (4.1)$$

The number of true positives over the sum of the number of true positive and the number of false negatives is known as Recall. So recall actually calculates the

number of the Actual Positives the machine learning model has captured by labeling it as Positive. The Recall is 74.96%.

$$R = \frac{|T_p|}{|T_p| + |F_n|} \quad (4.2)$$

F-measure is a function of precision and recall. The harmonic mean of precision and recall is known as F1-score. F1 score is a better evaluation measure if we want a balance between precision and recall and if that is an uneven class distribution. The F1 score is 75.21%.

$$F_1 = 2 \frac{P \times R}{P + R} \quad (4.3)$$

All values mentioned above are macro and weighted average³. In macro-average, values are taken from different sets and average is calculated on them. But in weighted-average, we consider difference in the number of samples in different sets.

4.3 User Study

We have conducted a user study. The task of the user study was to get feedback on the machine learning and visualization system. Users had to evaluate the topic classified by our system and were asked to provide feedback on the visualization system. They also answered a questionnaire on the visualization system and were evaluated on the time required to answer questions through the database and our visualization system. The data used for the machine learning system and the interactive visualization system is different in our user study.

4.3.1 Population

This user study was done on ten participants. The population for our user study was Graduate Students, Postdocs and Research Assistants in Computer Science at Dalhousie University who have read an academic research paper. That research paper can be their own published paper or a published paper that they have studied as part of their thesis or courses or just for interest.

³https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

4.3.2 User Tasks — Machine Learning System

Each user was asked to bring one research paper. The data used in this task is that research paper brought in by the user. This research paper is selected by the user and it aligns with their research interests. Hence, users are experts in the topic of this research paper. A user passes the title and abstract of their research paper through our machine learning system. This title and abstract is used by the machine learning system to predict the relevant topic(s). This set of classified topics are recommended to the user. Further, we asked the users to evaluate the predicted topics. User assesses our machine learning system by entering the number of topics that they think are correct out of all the predicted topics shown to them in the user interface. This helped us in testing the accuracy of the classifier/machine learning system from a user point of view.

4.3.3 User Tasks — Interactive Visualization System

The data used in this task is the ACM DSP data, which is the number of lectures on offer, number of lectures given, number of speakers, number of topics. This data is described briefly in Table A.1. Users were asked to answer a set of five questions. They have to answer these five questions using both the visualization system and the spreadsheet inspection. These questions are shown in Table B.1. These questions were asked, and the user was to answer them using either the visualization system connected to the ACM DSP database, or, alternatively, through the inspection of a set of spreadsheets extracted from the database to make the user interact with the visualization system. To remove bias, we asked the first five users to answer the questions first through our visualization system followed by the spreadsheet inspection and the last five users to do the same in reverse order. As part of the user study, we counted the time required for the users to search and answer the questions through both the methods, that is, spreadsheet inspection and our visualization system.

4.3.4 Results — Machine Learning System

Up to three topics were predicted for each title and abstract pair entered by the user. Each user selects a number from 0 to 3 to represent the number of correct topics predicted by the machine learning system as shown in Table 4.1. From the user study, we found that 24 topics were correct out of total 30 topics. Hence, on average, 80%

of the predicted topics were correctly predicted. The standard deviation is 0.8.

User No.	Number of Correct Topics
1	3
2	2
3	2
4	1
5	3
6	3
7	3
8	1
9	3
10	3
Average	2.4 (std: 0.8)

Table 4.1: No. of Correct Topics Predicted by the Machine Learning System as evaluated by the Users. Here second column refers to the number of correct topics that each user thinks are correct (chosen out of 3, which is the number of predicted topics from the machine learning system). std: Standard Deviation.

4.3.5 Results — Interactive Visualization System

The time required for the users to search and answer the questions through both the methods, that is, database and our visualization system, is shown in Table 4.2. From this user study, we found that the average time required for the users to answer the questions through the database is 291.1 seconds. However, the average time required for the same questions to be answered through our visualization system is 127.1 seconds. Hence our visualization system is 2.3 times faster than the time required to answer the same questions directly through the database or spreadsheet. Standard deviation of the all time values used by all users in the visualization system is 23.39 seconds. However, through the database/spreadsheet it is 30.60 seconds. Half of the difference between averages of the time required by users in the visualization system and by the database/spreadsheet is 82 seconds. Since the standard deviations are much smaller than half of the distance between the averages, so the difference of the averages is significant. We further performed the t-test on these numbers. The p-value was found to be less than 0.0001. Hence, this difference between averages is extremely significant.

User No.	Time with Visualization System	Time with Spreadsheets
1	120	288
2	115	305
3	122	321
4	124	291
5	121	281
6	183	346
7	104	244
8	107	252
9	159	318
10	116	265
Average	127.1 (std: 23.39)	291.1 (std: 30.60)

Table 4.2: Time spent (in seconds) by users in answering questions using the Visualization system and Database/Spreadsheet. std: Standard Deviation.

At the end of each user session, users answered the post-condition questionnaire as shown in Table 4.3. The purpose of this questionnaire was to get the overall user experience of systems. From this post-condition questionnaire we found that most users found the visualization system to be more interactive and interesting than the database. They have found the user interface of the visualization system to be more intuitive and easy to use as compared to the database. Moreover, some users have mentioned that they think that it is easy to classify a large set of research documents into topics using our machine learning system and they would like to use the system in the future. However, some users have mentioned in the feedback that they would like to see more options and filters available in the visualization system.

Question	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
The system has made it easy to classify a large set of research documents into topics.	0	0	1	2	7
The machine learning system is intuitive and easy to use.	0	0	0	3	7
The visualization system has made it simple to interact with the database.	0	0	0	1	9
The user interface of the visualization system is intuitive and easy to use.	0	0	0	1	9
The visualization system is fast enough while interacting with the database.	0	0	1	3	6
The number of options provided in the user interface for the visualization system are enough.	0	1	2	2	5
The overall system requires high technical skills.	8	0	1	0	1
With proper documentation, you want to use the software in future, if required?	0	1	1	1	7
How likely would you recommend this tool to anyone else, looking to classify a document into topics?	0	0	0	2	8

Table 4.3: Post-Condition Questionnaire (Completed by each user at the end of their session)

Chapter 5

Conclusion

This chapter mentions the overall conclusion of the thesis. Further possibilities of future work are also discussed in this chapter.

There are various methods to extract research articles for the keywords like Microsoft Academic API, Google Scholar, ArnetMiner, DBLP, Citeulike, and CiteSeer. The first thing that was analyzed during the research was that the data extracted using the Microsoft academic API is complete. The word complete here refers to complete title and abstract information for each keyword. Further, we have used three document representations which are bag of words, bag of categories, bag of concepts. We have also found that using the three different classifiers followed by the consensus methods has improved the performance of the machine learning model significantly. Further, our user study concluded that querying and answering through our visualization system is 2.3 times faster than doing the same through the database/spreadsheet and users found the system to be more interesting to use than a database/spreadsheet.

5.1 Future Work

Although the research study concluded many results, there are few possibilities of future work in it. The first possibility of improvement is in the data used for this research. ACM DSP committee has shared the data containing the list of topics delivered by the speakers. This data having topics is mapped to a list of keywords from the NSERC evaluation groups and research topics [10]. This task is being done manually and there are possibilities of improvement in this task. It can be automated by finding the similarity between the topics and the possible keywords.

Currently, we are using the Microsoft Academic API for searching and extracting the research articles corresponding to the keywords. But there are many other tools or APIs which exists and can be used as an alternative to the Microsoft Academic API. Some of those tools/APIs are Google Scholar, ArnetMiner, DBLP, Citeulike,

and CiteSeer. Hence some of these tools can be used to extract data instead of using the Microsoft Academic API.

The Helmholtz principle [1] states that whenever some large deviation from randomness occurs, a structure is perceived. This approach takes into account the document structure in order to enhance pure statistic summarization. Hence, after finding the concepts and categories in the given text, the Helmholtz principle can be applied to extract meaningful terms [17] from the bag of words, bag of concepts, and bag of categories.

Currently, the thesis is focusing only on the topics/categories related to Computer Science. But in the future work, this list of topics can be expanded to include other domains or fields as well. This can be achieved by choosing the keywords corresponding to the other topics and then following the same methods for further research. However, this may lead to a large number of classes. Hence, instead of predicting up to only three classes for a particular topic and abstract, we can predict all the classes along with their probabilities in decreasing order.

Currently, our machine learning system is predicting up to three topics using the model architecture explained in Fig. 3.1. However, there is a possibility of trying multi-label classification for this purpose. Multi-label classification [27, 9] is a classification problem where multiple target labels can be assigned to each observation instead of only one like in multi-class classification¹. This has not been tried in the current thesis but can be tried in the future work to check if this approach gives better results.

There are few possibilities of improvement in the visualization system as well. The feedback provided in the user study can be used to add more options and expand the system. In our user study, participants used title and abstract from only one academic research paper. However, this can be expanded by asking each user to bring multiple academic research papers to be used for testing the machine learning system.

The current machine learning model outputs up to three topics without probabilities. Hence this classification is not soft. Soft classifiers explicitly estimate the class conditional probabilities and then perform classification based on estimated probabilities. But, hard classifiers directly target on the classification decision boundary

¹<https://mlr.mlr-org.com/articles/tutorial/multilabel.html>

without producing the probability estimation [20]. This classification can be made soft by considering each consensus method to produce a ranked list of topics with associated probabilities. Further, these three ranked lists from three consensus methods can be combined into a single ranked list of topics. From this single list, top k topics can be chosen. Hence, the machine learning system can output k topics.

There is also some possibility of future research on the number of research paper (title and abstract pairs) extracted for each NSERC keyword using the Microsoft Academic API. Currently, we are extracting 1000 title and abstract pairs for each NSERC keyword and these are limited to last five years. However, this number can be changed and experimented with to find the possibility of having an optimal number which can produce better results.

For the visualization system, we have only 449 number of lectures offered, 61 number of lectures given and 124 number of speakers available to be used. This thesis focuses on finding all the lectures given in the past based on the lecture/event time, speaker name or lecture topic. This limited data prevents research on the time component in our thesis. However, the time component of this work can be expanded by requesting/gathering more data from ACMDSP committee on the number of lectures given over the entire lifetime of the ACMDSP. Past lecture times can be used to see the trend of lectures across the World. Further, data related to recruitment of new speakers and retirement of old speakers can also be requested from ACMDSP committee. Speaker location, time of recruitment, time of retirement can be used as features to predict the requirement of new speaker recruitment in a geographical area or country.

Bibliography

- [1] J.-M. Morel A. Desolneux, L. Moisan. *The Helmholtz Principle*, pages 31–45. Interdisciplinary Applied Mathematics. Springer New York, New York, NY, 2008.
- [2] Eneko Agirre, Ander Barrena, and Aitor Soroa. Studying the Wikipedia Hyperlink Graph for Relatedness and Disambiguation. *arXiv:1503.01655 [cs]*, March 2015. arXiv: 1503.01655.
- [3] Mohammad Al-Smadi, Omar Qawasmeh, Mahmoud Al-Ayyoub, Yaser Jararweh, and Brij Gupta. Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels reviews. *Journal of Computational Science*, 27:386–393, July 2018.
- [4] Xiao Cheng and Dan Roth. Relational Inference for Wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1787–1796, Seattle, Washington, USA, 2013.
- [5] Karthik Dinakar, Roi Reichart, and Henry Lieberman. Modeling the Detection of Textual Cyberbullying. In *Fifth International AAAI Conference on Weblogs and Social Media*, pages 11–17, Barcelona, Catalonia, Spain, July 2011.
- [6] Sarah Ellinger, Prantik Bhattacharyya, Preeti Bhargava, and Nemanja Spasojevic. Klout Topics for Modeling Interests and Expertise of Users Across Social Networks. *arXiv:1710.09824 [cs]*, Oct 2017. arXiv: 1710.09824.
- [7] Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, pages 1606–1611, Hyderabad, India, 2007. Morgan Kaufmann Publishers Inc.
- [8] Marcos Antonio Mourio Garca, Roberto Prez Rodriguez, and Luis E. Anido Rifn. Biomedical literature classification using encyclopedic knowledge: a Wikipedia-based bag-of-concepts approach. *PeerJ*, 3:e1279, September 2015. <https://doi.org/10.7717/peerj.1279>.
- [9] Eva Gibaja and Sebastin Ventura. A Tutorial on Multilabel Learning. *ACM Comput. Surv.*, 47:52:1–52:38, April 2015.
- [10] Natural Sciences and Engineering Research Council of Canada Government of Canada. NSERC List of Evaluation Groups and Research Topics: Computer science eg 1507. http://www.nserc-crsng.gc.ca/Professors-Professeurs/Grants-Subs/dgplist-psdliste_eng.asp#1507.

- [11] Anne-Wil Harzing and Satu Alakangas. Microsoft Academic: Is the phoenix getting wings? *Scientometrics*, 110:371–383, January 2017.
- [12] I-Han Hsiao and Piyush Awasthi. Topic Facet Modeling: Semantic Visual Analytics for Online Discussion Forums. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, LAK '15*, pages 231–235, Poughkeepsie, New York, 2015. ACM.
- [13] Sven E. Hug and Martin P. Brndle. The coverage of Microsoft Academic: analyzing the publication output of a university. *Scientometrics*, 113(3):1551–1571, December 2017.
- [14] Sven E. Hug, Michael Ochsner, and Martin P. Brndle. Citation analysis with microsoft academic. *Scientometrics*, 111:371–378, April 2017.
- [15] Carina Jacobi, Wouter van Atteveldt, and Kasper Welbers. Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, 4:89–106, January 2016.
- [16] David Kahle and Hadley Wickham. ggmap: Spatial Visualization with ggplot2. *The R Journal*, 5(1):144, 2013.
- [17] Dominik Krzemiski, Helen Balinsky, and Alexander Balinsky. Helmholtz Principle on Word Embeddings for Automatic Document Segmentation. In *Proceedings of the ACM Symposium on Document Engineering 2018*, pages 40:1–40:4, New York, NY, USA, 2018. ACM.
- [18] Hua Li, Daniel J. T. Powell, Mark Clark, Tifani O’Brien, and Rafael Alonso. User Modeling of Skills and Expertise from Resumes. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2015*, pages 229–233, Lisbon, Portugal, 2015. SCITEPRESS - Science and Technology Publications, Lda.
- [19] Marek Lipczak, Arash Koushkestani, and Evangelos Milios. Tulip: Lightweight Entity Recognition and Disambiguation Using Wikipedia-based Topic Centroids. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation, ERD '14*, pages 31–36, Gold Coast, Queensland, Australia, 2014. ACM.
- [20] Yufeng Liu, Hao Helen Zhang, and Yichao Wu. Hard or Soft Classification? Large-margin Unified Machines. *Journal of the American Statistical Association*, 106:166–177, March 2011.
- [21] David Milne and Ian H. Witten. An open-source toolkit for mining Wikipedia. *Artificial Intelligence*, 194:222–239, January 2013.

- [22] Afiz Momin. Towards Expertise Modeling Using Hierarchical Classification and Wikipedia Knowledge. Master's thesis, Dalhousie University, Faculty of Computer Science, Dec 2016. <https://DalSpace.library.dal.ca//handle/10222/72603>.
- [23] D. A. Ostrowski. Using latent dirichlet allocation for topic modelling in twitter. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pages 493–497, February 2015.
- [24] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.
- [25] Emmanuel Stefanakis and Kostas Patroumpas. *Google Earth and XML: Advanced Visualization and Publishing of Geographic Information*. Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [26] Mike Thelwall. Microsoft Academic automatic document searches: Accuracy for journal articles and suitability for citation analysis. *Journal of Informetrics*, 12(1):1–9, February 2018.
- [27] Grigorios Tsoumakas and Ioannis Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3:1–13, July 2007.
- [28] Christophe Van Gysel, Maarten de Rijke, and Marcel Worring. Unsupervised, Efficient and Semantic Expertise Retrieval. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, pages 1069–1079, Montréal, Qubec, Canada, 2016. International World Wide Web Conferences Steering Committee.
- [29] J. Zhang and H. Shi. Geospatial visualization using google maps: A case study on conference presenters. In *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*, pages 472–476, August 2007.
- [30] Y. Zhu. Introducing Google Chart Tools and Google Maps API in Data Visualization Courses. *IEEE Computer Graphics and Applications*, 32(6):6–9, November 2012.

Appendix A

Implementation Details

A.1 Data

This thesis work uses the data containing title and abstract information for a particular research article. This data is extracted using the Microsoft Academic API. Further, we have mainly three excel sheets with the names — speakers, lecture and topics data. The speaker sheet doesn't have location data for all speakers. Hence a crawler script was written to extract the speaker name and location from the ACM website. The location data for each speaker contains the city name, the state name, and the country name. This location data is used to fill any missing location data in the speaker excel sheet. Speaker excel sheet has many columns namely — Lecturer, Lecturer Email, City, State, Lecturer Country, Employer, Employer Type, Name of Nominator, Referred by, Number of Lectures, Last Updated, Accepted Date, Expire Date, Client No., Awards, Featured Speaker. The lecture excel sheet has lecture details. It mainly contains lecture location, speaker name who delivered the lecture, topic of the Lecture, event date, host organization. And the third sheet, which is topics excel sheet has two columns namely — topic and keywords. These sheets are used to make a database to use for our system. The code and complete data for the both, the machine learning system and visualization system, is added on github¹.

Data Component	Number
No. of Lectures on Offer	449
No. of Lectures Given	61
No. of Speakers	124
No. of Topics	42

Table A.1: Statistics of Data

¹<https://github.com/deepakmunjal15/visual-analytics-of-research-community-expertise>

Data size i.e. No. of Lectures on Offer, No. of Lectures Given, No. of Speakers and No. of Topics is shown in Table A.1.

A.1.1 Creating Microsoft Academic API Key to use Extractor tool

Microsoft Academic Extractor requires an API key to access and collect data. The process to create a key is available at Microsoft official website. This requires signing up for the account and requesting an API key.

Options available in the GUI are:

- **Microsoft Academic Key** — The key created above needs to be entered in the text box here. This key validates the user and gives access to the API to collect the data.
- **Keyword** — The keyword for which we need to collect the papers (data).
- **Strict Check** — When marked checked, it searches for papers matching the exact keyword. By default, it is left unchecked. When unchecked, it searches for all papers matching with any word in the keyword.
- **Number of Paper Limit** — This keeps an upper cap on the number of papers to be collected.
- **Date Range (YYYY-MM-DD)** — Date range for which the papers need to be searched. The Date should be entered in the form of YYYY-MM-DD.
- **Attributes Required** — It provides the option to choose from the four main attributes of a paper i.e. Title, Abstract, Author Name, Date. All 4 attributes are marked checked by default. The abstract is collected in the form of bag of words i.e. it contains a list of words along with their index/position in the abstract.

All text boxes available in the GUI are mandatory to be filled. After entering all options, Collect Data button needs to be clicked. Depending on the number of papers required, it may take from a few seconds to minutes or more to collect the data. If the research articles/required data is collected successfully, then a message, “Success”, “Data has been collected in the file name paper_data.csv”, is displayed to

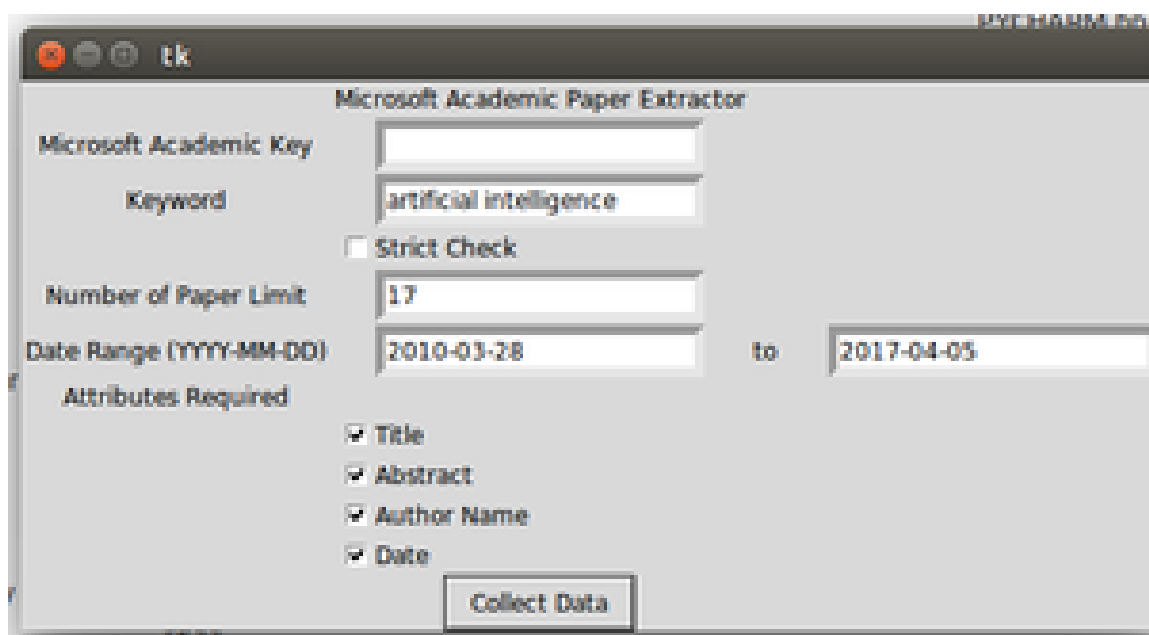


Figure A.1: Microsoft Academic Extractor Tool GUI

the user. If all text boxes are not filled before clicking on Collect Data button or if no relevant papers are found for the entered search options, then an error message, “Warning”, “Please enter correct input”, is displayed to the user. The GUI of the Microsoft Academic Extractor Tool is shown in Fig. A.1.

A.1.2 List of topics and keywords

We have the list of topics in the topics excel sheet. This data is shared by the ACMDSP Committee. But we have manually added a list of keywords corresponding to each topic using the NSERC research topics. The screenshot of the topics along with the list of keywords is shown in the below Fig. A.2.

A.1.3 Data Size

There are one hundred and twenty-four number of speakers. The list of speakers and their details are shown in Fig. A.3. Number of lectures on offer and number of lectures given are shown in Fig. A.4 and A.5 respectively.

There are 42 ACMDSP topics. Each topic is represented by on an average of 7 NSERC keywords. The overall number of NSERC keywords for all topics is 331. The list of ACMDSP topics and corresponding NSERC keywords are shown in Fig. A.2.

A	B	C	D
1 Topic	# of lectures for Topic		
2 Accessibility	11 computer accessibility	accessible computing	
3 Application Oriented Computing	40 mobile applications	E-health	
4 Architecture/Hardware	9 software architecture	computer architecture	
5 Artificial Language/Machine Learning	73 Knowledge representation and reasoning	machine learning	
6 Career-Related Topics	22 computing careers		
7 Cloud and Information Retrieval	37 security and privacy in the cloud	data management in the cloud	
8 Computer Graphics, Visualization and Interactive Techniques	45 Computer animation	geometric, procedural and volumetric modelling	
9 Computer Systems	71 reactive, embedded, and cyber-physical systems	Middleware	
10 Computers and Society	82 digital preservation	social issues in computing	
11 Data & Web Mining	1 educational data mining	business intelligence (information collection, data mining and analysis, business intelligence)	
12 Data Communication	16 data communications	information and communication theory	
13 Databases	0 database systems information	database security	
14 Databases & Information	26 information and communication theory	database systems information	
15 Design Automation	23 mechanism design	user interface design and evaluation	
16 Distributed Systems and Algorithms	19 Analysis of Algorithms for Distributed Systems	parallel and distributed algorithms	
17 Documentation	3		
18 Documentation	0		
19 E-Commerce	2 e-commerce		
20 Education	33 e-education	artificial intelligence and education	
21 Embedded Computer Systems	34 reactive, embedded, and cyber-physical systems	embedded systems	
22 Emerging Technologies	79		
23 Evolutionary Computation	0 evolutionary computation	Algorithms and computational genomics	
24 Game Development	11 computational game theory	computer game interfaces	
25 High Performance Computing	30 high-performance computing systems	architecture for high performance computing	
26 Human Computer Interaction	131 Usability engineering	user interface design and evaluation	
27 Knowledge Discovery in Data	48 Knowledge representation and reasoning	knowledge-based systems	
28 Management Information Systems	16 management information systems	Information systems: Models and principles	
29 Measurement & Evaluation	28 software engineering: evaluation	computing systems: performance models and evaluation	
30 Mobile Computing	54 mobile applications	HCI for mobile devices	
31 Multimedia	28 multimedia systems and networks	text and multimedia mining	

Figure A.2: List of ACM/DSP topics and corresponding NSERC keywords

A	B	C	D	E	F	
1 Lecturer	Lecturer Email	City	State	Lecturer Country	Employer	
2 Dr Nancy Amato	amato@cs.tamu.edu	College Station	Texas	USA	Texas A&M University	
3 Dr Saurabh Bagchi	sbagchi@purdue.edu	West Lafayette	Indiana	USA	Purdue University	
4 Dr David Bailey	david@davidhbailey.com	Davis	California	USA	Lawrence Berkeley National Laboratory (retired) and University of California	
5 Dr Brian Barsky	brian.barsky@sonic.net	Berkeley	California	USA	University of California, Berkeley	
6 Dr Kaveh Bazargan	kbazargan@acm.org	Tehran		Iran	Shahid Beheshti (National) University of Iran	
7 Dr Michel Beaudouin-Lafon	Michel.Beaudouin-Lafon@lri.fr	Paris		France	Université Paris-Sud (France)	
8 Dr Steve Benford	sdb@cs.nott.ac.uk	Nottingham	England	United Kingdom	The University of Nottingham	
9 Dr Regina Bernhaupt	Regina.Bernhaupt@irit.fr	Toulouse		France	IRIT	
10 Dr Indrajit Bhattacharya	indrajitb@gmail.com	Bangalore	Karnataka	India	IBM Research India	
11 Dr Shantanu Bhattacharya	shantanubhattacharya@yahoo.com	Canberra	ACT	Australia	Siemens Information Systems Bangalore	
12 Dr Ann Blandford	a.blandford@ucl.ac.uk	London		United Kingdom	UCL	
13 Dr Christophe Bobda	bobda@acm.org	Washington	DC	USA	CSCE	
14 Dr Stephen Brewster	Stephen.Brewster@glasgow.ac.uk	Glasgow		United Kingdom	University of Glasgow	
15 Dr Michael Bronstein	michael.bronstein@ust.ch	Lugano		Switzerland	University of Lugano / Intel Perceptual Computing	
16 Dr Erik Brunvand	eb@cs.utah.edu	Pittsburgh	Pennsylvania	USA	Carnegie Mellon University	
17 Dr Margaret Burnett	burnett@eecs.oregonstate.edu	Corvallis	Oregon	USA	Oregon State University	
18 Dr Tracy Camp	tcamp@mines.edu	Golden	Colorado	USA	Colorado School of Mines	
19 Dr Li Chen	lchen@udc.edu	Washington	DC	USA	University of the District of Columbia	
20 Dr Alain Chesnais	chesnais@acm.org	Toronto	Ontario	Canada	TrendSpottr	
21 Dr Keith Cheverst	k.cheverst@lancaster.ac.uk	Lancaster	Lancashire	United Kingdom	Lancaster University	
22 Dr Luca Chittaro	luca.chittaro@uniud.it	Udine		Italy	University of Udine	
23 Dr Elizabeth Churchill	churchill@acm.org	Mountain View	California	USA	Google	
24 Dr Donald Costello	dcostello@cse.unl.edu	Lincoln	Nebraska	USA	University of Nebraska	
25 Dr Gargi Dasgupta	gdasgupta@in.ibm.com	Bangalore	Karnataka	India	IBM Research	
26 Dr Dipankar Dasgupta	dasgupta@memphis.edu	Memphis	Tennessee	USA	The University of Memphis	
27 Dr Kerstin Dautenhahn	k.dautenhahn@herts.ac.uk		kerstin.dautenhahn@gmail.com	Hertfordshire	England	United Kingdom
28 Dr Gianluca Demartini	g.demartini@sheffield.ac.uk	Sheffield	England	United Kingdom	University of Sheffield	
29 Dr Tassos Dimitriou	tassos.dimitriou@ieee.org	Athens		Greece	Computer Technology Institute, Greece and Computer Eng. Dept., Kuwait U	
30 Dr Falko Dressler	dressler@ccs-labs.org	Paderborn		Germany	University of Paderborn	
31 Dr Henry Duh	Henry.Duh@utas.edu.au	Hobart	Tasmania	Australia	UTAS	

Figure A.3: List of speakers and their details

lecture_id	lecture_name	lecturer	related_topics
9223	The Big News of the 21st Century	Seth Shostak	Science & Computing;
9203	When Will We Find Extraterrestrial Life?	Seth Shostak	Science & Computing;
9364	"Telco Big Data: Current State & Future Directions"	Demetrios Zinailopo	Application Oriented Computing;Computer Graphics, Visualization
6804	3D Content For The Web	Alain Chesnais	Career-Related Topics;Computer Graphics, Visualization and Inter
9665	A Descriptive Analysis of Abnormal Stock Price Movement Following Financial News Article Release	Robert Schumaker	Artificial Language/Machine Learning;Knowledge Discovery in Data
8544	A Digital Socioscope	Ingmar Weber	Computers and Society;Human Computer Interaction;Knowledge E
7745	A Free Digital Society: What Makes Digital Inclusion Good Or Bad?	Richard Stallman	Computers and Society;Open Source;Operating Systems;Security
7186	A History of Computer Graphics	Craig Halpern	Computer Graphics, Visualization and Interactive Techniques;
7723	A Journey of Test Scripts: From Manual to Adaptive and Beyond	Mark Grechanik	Software Engineering;
2182	A Procedural Perspective	Craig Halpern	Computer Graphics, Visualization and Interactive Techniques;
5483	A Secure Data Aggregation based Trust Management Approach for Dealing with Untrustworthy Nodes in Sensor Network	Sanjay Kumar Madr	Cloud and Information Retrieval;Databases & Information;Mobile Co
5485	A Secure Data Sharing and Query Processing Framework via Federation of Cloud Computing	Sanjay Kumar Madr	Cloud and Information Retrieval;Databases & Information;Mobile Co
7765	A short history of Human Computer Interaction: a people-centred perspective	Geraldine Fitzpatrick	Human Computer Interaction;
7846	A Tale of Two Rendering Algorithms: Ray Tracing, Rasterization, and their Supporting Hardware	Erik Brunvand	Computer Graphics, Visualization and Interactive Techniques;Com
1943	A Transaction Model and Multiversion Concurrency Control For Mobile	Sanjay Kumar Madr	Databases & Information;Mobile Computing;Web Topics;
7263	A View of Usability and UX from the Viewpoint of User Engineering	Masaaki Kurosu	Management Information Systems;
4382	Accelerating Business Analytics Applications	Valentina Salapura	Computer Systems;High Performance Computing;
7605	Accelerating Data Discovery for Better Health	Laura Haas	Databases & Information;Knowledge Discovery in Data;
9323	Accelerating Deep Learning	Michael Gschwind	Artificial Language/Machine Learning;Computer Systems;High Per
7603	Accelerating the Discovery of Insights from Data	Laura Haas	Databases & Information;Knowledge Discovery in Data;
5867	Actionable Video Content Summarization: Lessons from Practical Case Studies	Sharath Pankanti	Accessibility;Artificial Language/Machine Learning;Cloud and Infor
9143	Adapting BDD for software maintenance projects using the FEep model.	Ranjith Tharayil	Application Oriented Computing;Management Information System;
9443	Adaptive Ability for Quality of Service	Theo Schlossnagle	Distributed Systems and Algorithms;
8403	Adaptive Multi-Factor Authentication (A-MFA) Methodology	DipanKar Dasgupta	Cloud and Information Retrieval;Computer Systems;Emerging Tech
8127	Adaptive Simulation with Triangle Meshes	James O'Brien	Computer Graphics, Visualization and Interactive Techniques;Gam
7967	Adding More Than Two Dimensions to Tablet Interfaces: Is Tony Stark home?	Joaquim Jorge	Computer Graphics, Visualization and Interactive Techniques;Mult
7324	Advanced Interaction for Information Visualization	Jean-Daniel Fekete	Computer Graphics, Visualization and Interactive Techniques;Hum
7987	Advancements in Intelligent Support for Collaborative Learning	Seiji Isotani	Application Oriented Computing;Education;Human Computer Inter

Figure A.4: No. of Lectures on Offer

speaker	req_date	org_name	org_type
Jennifer Golbeck	07/06/2017	ACM User Modelling, Adaptation and Personalization 2017	conference (ACM User Modelling, Adaptation and Personalization 2017)
Pearl Pu	04/06/2017	UMAP 2017 - User Modelling, Adaptation and Personalization (UMAP 2017)	conference (UMAP 2017 - User Modelling, Adaptation and Personalization (UMAP 2017))
Yung-Hsiang Lu	03/09/2017	Chongqing University of Posts and Telecommunications	conference (Chongqing University of Posts and Telecommunications)
Toby Walsh	05/18/2017	Chinese Association for Artificial Intelligence	Professional_Group_not_affiliated_with_ACM (Chinese Association for Artificial Intelligence)
Toby Walsh	07/03/2017	Society for Out-of-Frame Education	Professional_Group_not_affiliated_with_ACM (Society for Out-of-Frame Education)
Abhik Roychoudhury	03/20/2017	The world's largest technical professional organization for the advancement of technology	ieee_chapter (The world's largest technical professional organization for the advancement of technology)
Valentina Salapura	09/15/2016	The Science and Information Organization	conference (The Science and Information Organization)
Tracy Camp	07/08/2017	SRM UNIVERSITY	ACM_chapter (SRM UNIVERSITY)
Beverly May	05/08/2017	San Francisco Bay Area Chapter of ACM	ACM_chapter (San Francisco Bay Area Chapter of ACM)
Panagiota Fatourou	03/03/2017	West University of Timisoara (Universitatea de Vest din Timisoara)	conference (West University of Timisoara (Universitatea de Vest din Timisoara))
Indrajit Bhattacharya	05/14/2017	Marathwada Mitra Mandali's College of Engineering, Pune	ACM_chapter (Marathwada Mitra Mandali's College of Engineering, Pune)
Scott Jenson	03/09/2017	Mexican Interaction Lab MILab	ACM_chapter (Mexican Interaction Lab MILab)
Albert Bifet	06/20/2017	Universidade Federal de Uberlândia	conference (Universidade Federal de Uberlândia)
Jean Vanderdonck	05/05/2017	University of Rostock	University_not_affiliated_with_ACM_chapter (University of Rostock)
Geraldine Fitzpatrick	06/18/2017	ACM Intl. Conference on Interactive Surfaces and Spaces, ISS 2017	conference (ACM Intl. Conference on Interactive Surfaces and Spaces, ISS 2017)
Gargi Dasgupta	09/30/2017	SRM University Kattankulathur Campus	ACM_chapter (SRM University Kattankulathur Campus)
Joaquim Jorge	03/27/2017	Instituto de Computação da Universidade Federal Fluminense (IC-UFF)	conference (Instituto de Computação da Universidade Federal Fluminense (IC-UFF))
Elien Voorhees	08/28/2017	Gettysburg College	ACM_chapter (Gettysburg College)
Shrisha Rao	07/18/2017	University of Engineering and Management, Jaipur	ACM_chapter (University of Engineering and Management, Jaipur)
Oge Marques	09/05/2017	pontifical catholic university of parana	conference (pontifical catholic university of parana)
Fernando Koch	08/03/2017	Bina Nusantara University	University_not_affiliated_with_ACM_chapter (Bina Nusantara University)
Bebo White	08/13/2017	J. Boye	conference (J. Boye)
Margaret Burnett	05/17/2017	Simon Fraser University, School of Computing Science	University_not_affiliated_with_ACM_chapter (Simon Fraser University, School of Computing Science)
Ponnurangam Kumaraguru	10/13/2017	Shiv Nadar University	ACM_chapter (Shiv Nadar University)
Panagiota Fatourou	09/22/2017	ACM-W Uppsala student chapter	ACM_chapter (ACM-W Uppsala student chapter)
James Hendler	05/13/2017	The Science and Information Organization	conference (The Science and Information Organization)
Michael Gschwind	08/17/2017	FEU Institute of Technology (FEU East Asia College)	ACM_chapter (FEU Institute of Technology (FEU East Asia College))
Pearl Pu	09/07/2017	School of Computing, University of Leeds	University_not_affiliated_with_ACM_chapter (School of Computing, University of Leeds)
Gabriel Loh	11/09/2017	ISCTE-IUL Instituto Universitario de Lisboa	ACM_chapter (ISCTE-IUL Instituto Universitario de Lisboa)
Albert Bifet	08/25/2017	Ministry of Communication and Information Technology	conference (Ministry of Communication and Information Technology)

Figure A.5: No. of Lectures Given

For each keyword, we have extracted 1000 title and abstract pairs using the Microsoft Academic API. The list of title, abstract and ACMDSPTOPICS is shown in Fig. A.6.

A.2 NSERC Research Topics and Keywords

NSERC is the Natural Sciences and Engineering Research Council of Canada. NSERC has embraced a Conference model for the audit of Discovery Grant applications. This model was embraced dependent on the suggestions from the International Review of Discovery Grants and the Grant Selection Committee (GSC) Structure Review. NSERC replaced the Grant Selection Committees with the twelve discipline-based

tensorflow a system for large scale machine learning	{u'operations': [26], u'represent': [20], u'project.': [133], u'focus': [110], u'networks.': [118], u'incluing': [53], Artificial Language/Machine Learning
distilling the knowledge in a neural network	{u'show': [112], u'results': [107], u'we': [103, 140], u'rapidly': [181], u'knowledge': [78, 130], u'heavily': [123], Artificial Language/Machine Learning
learning deconvolution network for semantic segmentation	{u'all': [71], u'semantic': [4, 63], u'proposed': [78], u'among': [131], u'masks.': [46], u'labels': [42], u'results': [6], Artificial Language/Machine Learning
on the properties of neural machine translation encoder decoder approach	{u'and': [27, 41, 70, 106], u'network.': [78], u'representation.': [50], u'often': [22], u'proposed': [73, 118], u'should': [118], Artificial Language/Machine Learning
deep pose human pose estimation via deep neural networks	{u'and': [56], u'detailed': [77], u'formulated': [18], u'art': [82], u'simple': [59], u'is': [17], u'in': [38, 52, 70], u'po': [118], Artificial Language/Machine Learning
spatial transformer networks	{u'limited': [13], u'show': [96], u'lack': [16], u'results': [103], u'existing': [64], u'translation.': [110], u'still': [12], Artificial Language/Machine Learning
faster r-cnn towards real time object detection with region proposal network	{u'all': [129], u'per': [153], u'computation': [33], u'optimization.': [100], u'networks': [3], u'objectness': [74], Artificial Language/Machine Learning
big data and machine learning to revamp computational toxicology and informatics	{u'Relationships': [70], u'selection': [108], u'interpretation': [104], u'learning-based': [115], u'incluing': [100], Artificial Language/Machine Learning
asynchronous methods for deep reinforcement learning	{u'and': [5, 34], u'instead': [83], u'all': [46], u'have': [39], u'show': [35, 89], u'simple': [4], u'successfully': [50], Artificial Language/Machine Learning
a fast and accurate dependency parser using neural networks	{u'all': [1], u'just': [58], u'generalize': [18], u'computation': [25], u'speed': [28], u'indicator': [11], u'based': [6], Artificial Language/Machine Learning
deepwalk online learning of social representations	{u'all': [151], u'show': [102], u'able': [148], u'random': [64], u'results': [101], u'statistical': [31], u'world': [192], Artificial Language/Machine Learning
photo realistic single image super resolution using a generative adversarial network	{u'manifold': [163], u'similarity': [191, 194], u'able': [203], u'minimizing': [61], u'consists': [144], u'fidelity': [95], Artificial Language/Machine Learning
learning spatiotemporal features with 3d convolutional networks	{u'spatiotemporal': [8, 37], u'compact': [103], u'all': [54], u'networks': [15], u'features': [101], u'layers': [55], Artificial Language/Machine Learning
deep generative image models using a laplacian pyramid of adversarial networks	{u'the': [42, 52, 90, 111, 116], u'pyramid': [29], u'real': [85], u'samples': [59], u'from': [61, 98, 107], u'is': [49], Artificial Language/Machine Learning
deeply supervised nets	{u'focus': [25], u'nets': [3], u'layers': [42], u'we': [0, 24, 98], u'to': [82], u'stochastic': [107], u'hidden': [20, 78], Artificial Language/Machine Learning
draw a recurrent neural network for image generation	{u'eye.': [84], u'art': [55], u'iterative': [41], u'and.': [61], u'foveation': [26], u'variational': [34], u'image': [13], Artificial Language/Machine Learning
hypercolumns for object segmentation and fine grained localization	{u'precise': [32, 41], u'all': [69], u'show': [81, 122], u'over': [115, 127], u'[22]': [92], u'6.6': [124], u'networks': [5], Artificial Language/Machine Learning
an evolutionary many objective optimization algorithm using reference points	{u'all': [158], u'evolutionary': [7, 33], u'multi-objective': [2, 34], u'developed': [1], u'non-dominated': [95], Artificial Language/Machine Learning
recurrent neural network regularization	{u'and': [32, 46, 68], u'tasks': [59], u'translation.': [70], u'show': [38, 47], u'simple': [3], u'image': [65], u'Recurrent': [118], Artificial Language/Machine Learning
end to end memory networks	{u'recurrent': [7], u'less': [40, 111], u'over': [10], u'both': [133], u'competitive': [105], u'tasks': [87], u'we': [0], Artificial Language/Machine Learning

Figure A.6: Training Data — Title, Abstract and corresponding ACM/DSP Topic

Evaluation Groups. Applicants to the Discovery Grants Program are asked to recommend an Evaluation Group and a Research Topic(s) that best fits the subject of their proposal. These disciplines are Genes, Cells and Molecules; Biological Systems and Functions; Evolution and Ecology; Chemistry; Physics; Geosciences; Computer Science; Mathematics and Statistics; Civil, Industrial and Systems Engineering; Electrical and Computer Engineering; Materials and Chemical Engineering; Mechanical Engineering.

NSERC has 12 Evaluation Groups. Each Evaluation Group has multiple Research Topics. There are 21 NSERC Research Topics in the Computer Science Evaluation Group². Each Research Topic has a set of Keywords. The total number of NSERC keywords for Computer Science Evaluation Group is 296. The list of NSERC Research Topics and the corresponding Keywords is shown in Table A.2.

1507 Computer Science		
Label	Research Topic	List of Keywords
CS01	Web-Enabled Applications and Services (E-*)	E-health; e-business; e-government; e-learning; e-commerce; e-culture; e-education; e-science; mobile applications
CS02	User Adaptive Systems	User modelling; mechanism design; user adaptive interaction; artificial intelligence and education; adaptive learning systems; educational data mining

²http://www.nserc-crsng.gc.ca/Professors-Professeurs/Grants-Subs/dgplist-psdliste_eng.asp#1507

CS03	Mathematical Computing	Symbolic computing; scientific computing; numerical optimization; computer algebra; numerical modelling and simulation
CS04	Theory of Computing	Theoretical foundations of computation; complexity theory; structural complexity; logic and proof complexity; descriptive complexity; automata theory; information theory; coding theory
CS05	Algorithms and Data Structures	Analysis of algorithms; data structures; parallel and distributed algorithms; graph algorithms; computational combinatorics; computational geometry; randomized algorithms; computational game theory; theoretical cryptography.
CS06	Computer Networks	Network protocols; protocol performance; data communications; simulation and emulation of networks; multimedia systems and networks; network management; wireless and mobile networks and ad hoc networks; sensor networks; optical networks; overlay networks and peer to peer networks; information and communication theory; network algorithms; pervasive computing (ubiquitous computing); green networks; cognitive networks; protocol testing.
CS07	Quantum Computing	Quantum complexity; quantum cryptography; quantum algorithms; quantum devices; quantum information, models of quantum computation; quantum coding

CS08	Information Systems	Models and principles; database systems information; storage and retrieval; information systems; information interfaces and presentations; information integration; visual data analysis; geographic information systems; management information systems; decision support systems; health information systems; medical informatics
CS09	Security and Privacy	Authentication, authorization and access control; anonymity and privacy; information and application-level security; biometrics; cryptographic protocols; database security; denial of service; intrusion detection and prevention; formal methods for security; formal models and provable security; network security; operating system security; language-based security; malicious code detection and emerging threats; design and verification of cryptographic protocols; privacy requirements; privacy policies; privacy preference languages; language-based privacy; private data management; privacy in Web services and semantic Web; secure multi-party protocols; cryptographic algorithms; security and privacy auditing; security and privacy in the cloud; security metrics, information flow control

CS10	Data Management	<p>Database management; text management; information organization and retrieval; semi- and unstructured data; managing uncertain information; data warehouses; business intelligence (information collection, data mining and analysis, business activity monitoring, business process management, decision analysis); digital libraries; community information management; stream data management; caching; data cleaning; text and multimedia mining; information extraction from text; sentiment analysis; prediction; clustering; graph and network analysis; temporal and sequence data mining; collaborative filtering and social information sharing; model complexity; quality metrics; spatial databases; temporal databases; scientific databases; data mining; data security and privacy; data management in the cloud; autonomic data management; mobile data management; distributed data management</p>
CS11	Programming Languages	<p>Compilers; semantics; type systems; semantic analysis; static analysis; programming paradigms; programming techniques; imperative programming; object-oriented programming; logic programming; functional programming; concurrent programming; event-driven programming; scripting languages; generative programming, domain-specific languages; modelling languages (semantics of, compilers for); multi-paradigm programming/modelling; dynamic analysis</p>

CS12	Software Engineering	Requirements; specification; software design; software architecture; software implementation; quality management-testing; validation; verification; software development environments; software analysis; evaluation; reliability; maintenance; user interface development; re-engineering and migration; user interfaces; software evolution; process life-cycle models; agile methods; model-driven development; reactive, embedded, and cyber-physical systems; software product lines; data mining from software repositories
CS13	Formal Methods	Verification; state and logic models; temporal logic; model checking; theorem proving; refinement; testing; semantics; formal languages; formal specification notations and languages; run-time monitoring; discrete event systems; synthesis and correctness-by-construction
CS14	Computing Systems	Middleware; architecture; real-time systems; embedded systems; operating systems; file and storage systems; input and output architectures; high-performance computing systems; system reliability and fault tolerance; virtualization; peer-to-peer systems; virtual machines; power management; cache; memory; green computing; enterprise systems; performance models and evaluation; reconfigurable computing; programmable matter

CS15	Parallel and Distributed Computing	Distributed models and algorithms; distributed architectures; distributed and parallel programming and languages; design, validation and verification; distributed storage; file systems; management; fault tolerance; performance analysis; parallelism and concurrency; parallel processing; parallel models and algorithms; high-performance computing; clusters; symmetric multi-processors; applications; peer-to-peer; grid computing; pervasive computing; map-reduce paradigm; cloud computing; multi-core architectures; service-oriented computing
CS16	Web-Based Systems	Social computing; social media; social networks; internet theory; Web services; standards; Web architectural styles (e.g., REST); design of Web systems; Web security; portals and portal frameworks; wikis; blogs; crowdsourcing; recommender systems
CS17	Human Computer Interaction	Usability engineering; user interface design and evaluation; multi-modal user interaction; computer-supported cooperative work; haptics; HCI in visualization; virtual reality; human-robot interaction; computer game interfaces; entertainment computing; mixed reality; HCI for mobile devices, modelling/simulation/synthesis of user interfaces

CS18	Artificial Intelligence	Intelligence	Knowledge representation and reasoning; machine learning; natural language processing (understanding); natural language generation; machine translation; evolutionary computation; genetic algorithms; genetic programming; cognitive science; cognitive modelling; intelligent agents; knowledge-based systems; constraints and search; planning and scheduling; agent-based and multi-agent systems; reasoning under uncertainty and sequential decision making; semantic Web; ontologies; speech understanding; artificial intelligence in games; neural networks
CS19	Computer Graphics and Visualization	Graphics and Visualization	Computer animation; geometric, procedural and volumetric modelling; rendering; visualization; interactive techniques; computational photography
CS20	Bioinformatics and Bioinspired Computing	Bioinformatics and Bioinspired Computing	Algorithms and computational genomics; data management, integration and visualization; software and database systems; computational biomodelling; computational neuroscience; DNA computing; molecular and atomic computing; evolutionary computation; genetic algorithms; genetic programming; neural networks
CS21	Computer Vision and Robotics	Vision and Robotics	Image understanding; computer vision; document image analysis; image processing; robotics; video analysis; medical image analysis

Table A.2: NSERC Research Topics and the corresponding Keywords for the Computer Science Evaluation Group

Appendix B

User Study Details

B.1 Introduction

The outcome of the thesis is evaluated through a user study. An application to conduct the user study was sent to the Social Sciences and Humanities Research Ethics Board for review.

B.1.1 Study Population and Plan

The research study had two sub-parts. The first sub-part of the study deals with the machine learning system. In this, users were asked to enter the title and abstract of an academic research paper in the machine learning system. The classifier/system categorizes it into one or more topics. These classified topics are chosen from a set of forty-two topics that we have defined for our machine learning system. This set of classified topics are shown to the user in the user interface. The user was asked to evaluate these topics and enter the number of correct topics. This helped us in testing the accuracy of the classifier/machine learning system from a user point of view.

The second sub-part of the study deals with the visualization system. These two subparts are in continuation and user did it in a single flow. A connection to the ACM DSP database was provided with a UI (User Interface). The users interacted with the various options provided within the UI to query the database.

This user study was done with ten users. The population was Graduate students, Postdocs and Research Assistants in Computer Science at Dalhousie University who have read an academic research paper. That research paper could have been their own published paper or a published paper that they have studied as part of their thesis or courses or just for interest. After completing the study, participants filled out a questionnaire to further express their views of the tool. This user study was

very important given the fact that testing a tool is an important part of a software development life cycle and provides useful feedback for future improvements.

The system designed as a part of the thesis automatically classifies a research paper into topics/classes using machine learning. But the automatic approaches are not always correct. Similarly, for the visualization system, there's a need for the user evaluation which will help in gathering feedback and improving the user interface further. Hence we needed user involvement to evaluate these systems, which could have been possible through a user study. This study helped us to get user feedback about the work. The outcome of the study reflected on how good the systems are as per user evaluation. This feedback will be very useful in improving the machine learning and visualization system further in future work.

B.1.2 Research Question

Our hypothesis for the user study was that it is faster and more accurate to interact with a database with our tool than with the excel sheets. The user's input took two forms: It included the user's interaction with the visualization system. This included clicking on a button, entering a search query in the search box. It also included the research paper, provided by the user, to pass its title and abstract through the machine learning system/classifier. This was also done directly using the user interface.

B.1.3 User Recruitment

Principal Investigator (PI), Deepak Munjal, sent an e-mail to all Graduate students, Postdocs and Research Assistants in Computer Science at csgrads@cs.dal.ca. Recruitment paper posters were also posted on notice boards in the Goldberg Computer Science building. Interested candidates could read the email or posters, read consent form by opening the link available in both recruitment email and recruitment poster. Interested candidates sent an email to PI on the email mentioned in both email and poster. They needed to send their Degree name and year of study in the email. Selection of ten users, from the pool of candidates who were interested in taking part in the study was based on the seniority level (Degree name and Year of study). Further, PI replied to them and invited them for the study at a date and time based on their availability.

B.1.4 Informed Consent Process

The participants were presented with the informed consent form at the beginning of the study. They needed to provide consent at the beginning of the study before they proceeded with the introduction and training. The consent was provided by choosing Yes from the radio button Yes or No. The link to the consent form was also included as part of the recruitment email and poster. This made it possible for participants to read the consent form prior to planning to be a part of the study. The consent form contained a brief introduction of the study, information about the confidentiality and anonymity of the participant's data, the participant's right to withdraw and the compensation. The online form was provided and administered by the PI at the very beginning of the study. The participants were mentioned in the beginning that they can withdraw from the study at any time they want without penalty. This was also mentioned in the consent form.

B.1.5 Study Design

We provided a quiet room in the Computer Science Faculty for the study. Participants were asked to perform the following tasks in 65 minutes:

- Participants read the consent form in our system's user interface and provided consent by choosing Yes radio button. This step took 10 minutes.
- After providing consent, participants were given several short examples to become familiar with the user interface and the whole system. This step also took 10 minutes.
- Each participant was given one user ID, which was used for testing both machine learning and visualization system. The post-condition (evaluation) questionnaire was given to the users as well. Before the visualization system test, users were provided with a set of questions as shown in Table B.1 that made users interact with the system.
- The participants were asked to perform document classification on the academic research paper that they brought with them. Then they evaluated the classified topics and wrote the number of correct topics in the user interface of our system

S.No.	Question
1	You are staying in New York and looking for a Speaker who can deliver a lecture on Artificial Language/Machine Learning in New York itself. Considering the limited budget, whom would you prefer?
2	How many Computer Graphics speakers need to be recruited in Germany?
3	Which topic is trending more from July 2017 to March 2018 in Canada?
4	Which country has the maximum number of seminars on Human-Computer Interaction in the year 2017?
5	How many seminars occurred on Text mining in New York in the year 2017?

Table B.1: Visualization System Test Quizzes (for making users interact with the visualization System and Spreadsheets)

itself. Participants spend about 10 minutes for this step. Similarly, for the visualization system, users spend about 20 minutes. However, the system was available for any participant who wanted to spend more time.

- After finishing the user study, the participant filled the evaluation questionnaire, submitted it, and received the compensation, which took 15 minutes.
- Hence, providing consent, introduction, and training took 20 minutes. Filling out the evaluation questionnaire took around 15 minutes. The system testing for both visualization system and machine learning system took about 30 minutes; however, participants were given ample time to do it so that they could spend more time with the system/interface.

B.1.6 Data Analysis

For machine learning system, data analysis focused on how many predicted topics users thought were correct. This number was collected and helped us in providing an estimate of the accuracy of the system. This will be very useful in improving the machine learning system. For visualization system, time that users have interacted with the database/excel sheets and time they have used to interact with the user interface of the visualization system was collected. This data was analyzed to see the total time required and to access the easiness of the user interface of the visualization system over accessing the database/excel sheets directly.