TOWARDS EXPERTISE MODELING USING HIERARCHICAL CLASSIFICATION AND WIKIPEDIA KNOWLEDGE

by

Afiz Momin

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

at

Dalhousie University
Halifax, Nova Scotia
December 2016

To Mom (Naseem Momin), Dad (Alimohmed Momin), Brother (Akil Momin) and my Wife (Muniza Maredia)

Table of Contents

List of Tables

List of Figures

Abstract

We define expertise modeling as profiling an expert, a knowledgeable person in one or more domains, based on evidence from research articles into one or more research topics. The traditional text classification approach involves classifying a document into a class where classification hierarchy is limited to one level. However, the real-world problems are more complex and could be related to hierarchical structure and therefore, there has been numerous research in a hierarchical classification. Millions of enthusiastic researchers contribute in the form of research articles in conferences or journal publications and apply for research grants, and the task of assigning reviewers to research articles and correct research topic for the grant application is non-trivial.

For our research, we have trained a hierarchical classifier on titles and abstracts of research articles and it predicts one or more research topics for a given article of an expert. We have used traditional Bag-of-Words (BOW) representations of the text which is enriched using a semantic knowledge from Wikipedia's concepts (BOC) and categories (BOK). For each of these document representations, a hierarchical classifier is trained and their outputs are combined using consensus methods to predict a research topic. In reality, research articles can belong to multiple research topics and therefore two approaches to multi-label a research article are proposed.

We evaluate and compare the performance of the hierarchical model with a baseline, a flat classifier, and using different training set and different evaluation measures such as precision, recall, and f-measure. The combined outputs from hierarchical classifiers, BOW, BOC, and BOK, are compared with a flat classifier and a hierarchical classifier based on BOW. The results from various approaches, comparison of the performance of different hierarchical classifiers and current issues are also discussed.

List of Abbreviations Used

BOC Bag-Of-Concepts

BOK Bag-Of-Categories

BOW Bag-Of-Words

CCQ Concept Chain Queries

ESA Explicit Semantic Analysis

LCL Local Classifier per Level

LCN Local Classifier per Node

LCPN Local Classifier per Parent Node

NSERC Natural Sciences and Engineering Research Council

O-v-R One-vs-Rest Classifier

t-SNE t-Distributed Stochastic Neighbor Embedding

VSM Vector Space Model

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr. Evangelos Milios, for his constant guidance, encouragement, and professionalism. It was a great opportunity to work under your aegis. Thanks for trusting me and giving an opportunity to learn and apply my skills. Many thanks for accepting my application for graduate research on text analytics which welcomed my greatest decision to join Dalhousie University and choosing Master's program. I would like to thank Dr. Seyednaser Nourashrafeddin for his support, timely guidance, for extending help whenever needed and for being a good partner during the research. I am thankful to Ehsan Sherkat who helped me with the setup of Wikipedia tools that saved a couple of weeks of time. I am truly grateful for their unwavering support.

With a graduate research funding from NSERC Engage Grant with Proximify Inc, it helped me to stay focused on my research work and the schedule. I would like to express my gratitude to NSERC committee for providing a grant for this work.

I would 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 during the journey of the graduate program.

Chapter 1

Introduction

Expertise modeling is about profiling an expert, a knowledgeable person in one or more domains, based on evidence from research articles into one or more research topics. Traditionally, finding an expert was manually achieved by interviews, by assessing the depth of the knowledge in the research areas and based on expert's self-assessment. However, this process is often erroneous and time-consuming. Academic institutes have profound research environment where researchers continuously publish new knowledge in journals, conferences, and personal blogs. Each of these resources is linked with authors, affiliations, citations, and publications. It altogether contributes to the academic network. In recent years, there has been increase in mining opportunities for analyzing plethora of academic corpus for various purposes such as to find an expert in particular domain, find correct class of the research article, create network within and outside organization based on expertise of researcher/group that enables senior administrator understand depth and breadth of research and future collaboration, visualize academic research growth, and in future governmentś Research & Development funding options.

All these aforementioned applications will become a nontrivial task as research community grows. Hence, a reliable system to identify correct expertise of researchers that change over a time and categorization to a common nomenclature is required.

The main objective of the research is to predict a research topic of a research paper as shown in Figure 1.1. The research topics are research groups of NSERC[1] evaluation groups. A hierarchical structure consists of evaluation groups (Table 1.1) at the first level and research topics (Table 1.2) as the second level or the leaf-level as shown in Figure 1.2. The computational model is a hierarchical classifier which

[1]http://www.nserc-crsng.gc.ca/Professors-Professeurs/Grants-Subs/DGPList-PSDListe_eng.asp

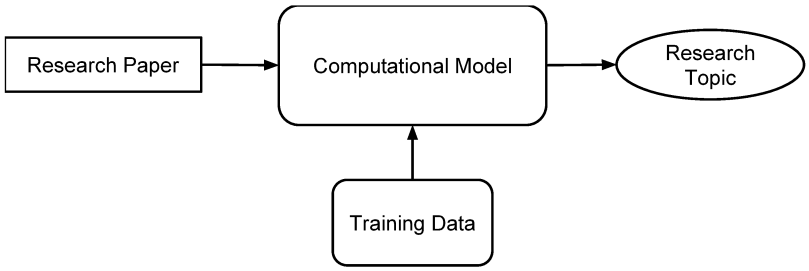


Figure 1.1: A trained computational model predict a research topic for a given research paper.

is trained on research articles extracted from well-known journals. The objective of the model is to identify for a given document d and the set of research topics and evaluation groups as classes C, a classes $J \subseteq C$ at each level and moves further down the hierarchy to predict one or more research topics. Our objective is to predict a research topic for a given research article.

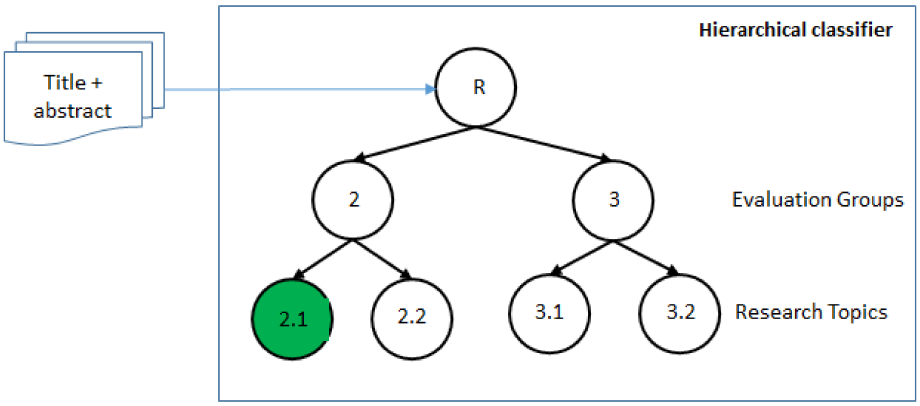


Figure 1.2: A computation model, a hierarchical classifier, classifies a research paper as a title and an abstract into research topic. R denotes the root node, evaluation groups the first level, and research topics the second.

We define expertise modeling as profiling an expert, a knowledgeable person in one or more domains, based on evidence from research articles into one or more research topics. In this thesis, we are focused on classifying a document into one or more research topics. The output from our proposed methodology can be used in various ways to profile experts. The term(s) document, research article(s), a title and an abstract and research paper(s) are used interchangeably in the thesis.

Id	Evaluation Groups
1501	Genes, Cells and Molecules
1502	Biological Systems and Functions
1503	Evolution and Ecology
1504	Chemistry
1505	Physics
1506	Geosciences
1507	Computer Science
1508	Mathematics and Statistics
1509	Civil, Industrial and Systems Engineering
1510	Electrical and Computer Engineering
1511	Materials and Chemical Engineering
1512	Mechanical Engineering

Table 1.1: Evaluation groups defined by NSERC

The high level process to create a training data to train a computation model is explained in Figure 1.3. A journal's aims and scope section uses keywords to define the scope which are compared with each research topic in an evaluation group. The research topic with exclusive match in an evaluation group is retrieved. Using journal's ISSN number the articles of the journals are retrieved and each of these articles are labeled with research topic label (refer Table 1.2).

Genes, Cells and Molecules		
Label	Research Topic	Keywords
LSA01	Immunology	Host-cell interactions; immune response; antigens; antibodies; host-pathogen interactions; immunogenetics; innate immunity; cytokines and antimicrobials; antigen presentation; inflammation; lymphocyte; neutrophil; monocyte; macrophage; sinus; thymus epithelium; lymph node; spleen; chemokine; interleukin; dendritic cell; B cell; T cell; plasma cell; mucosal immunity; immunoglobin; ecological immunology; Toll-like receptors; evolution of immune responses

Table 1.2: An example of a research topic and keywords from evaluation group LSA (1501) defined by NSERC.

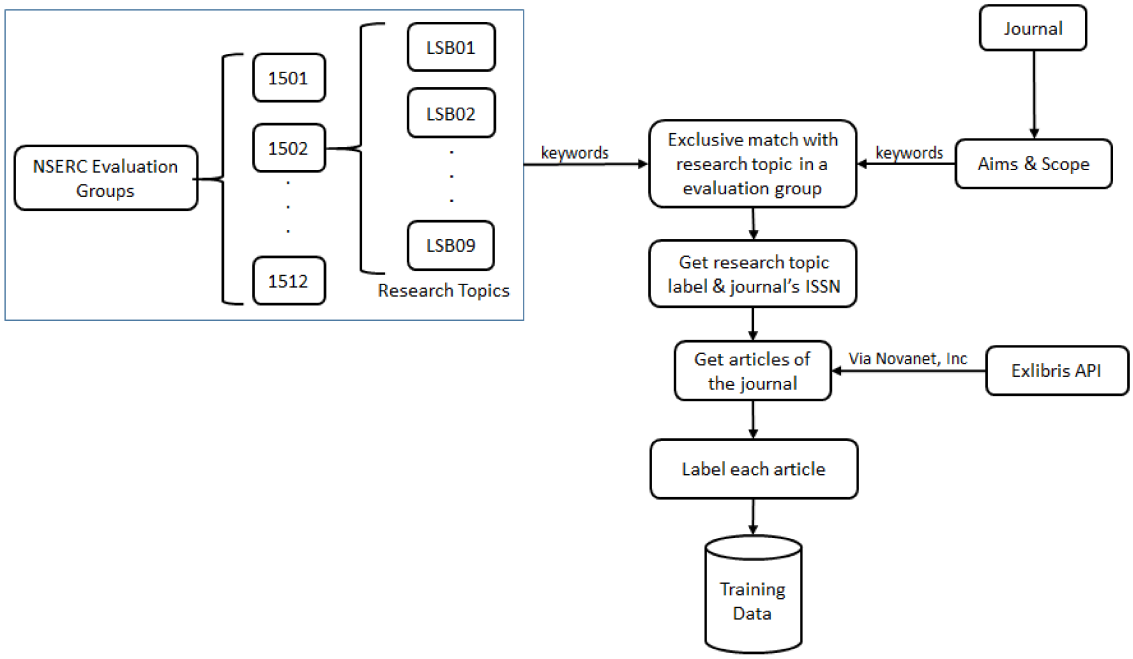


Figure 1.3: High level overview of creating a training set for hierarchical classifier.

A hierarchical classifier is trained on raw input of titles and abstracts from journals whose aim and scope matches keywords defined by Natural Sciences and Engineering Research Council (NSERC) for each research topic. Three different representations for each text based on lexical and semantic statistics are created. For each of these representations, a hierarchical classifier is trained. For a given scientific paper, we feed its title and abstract into hierarchical classifiers to predict research topics which are inputs to consensus methods to predict a single research topic as shown in Figure 1.4.

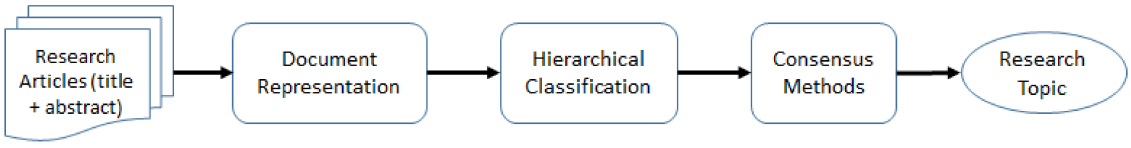


Figure 1.4: High-level overview of the methodology.

The methodology is then tested using research articles of each expert and a profile is created. The profile is the list of research topics predicted by hierarchical classifiers

and consensus methods. The research topics in each profile could be ranked based of number of occurrences in the predicted output to create a ranked profile. A general test approach for our methodology is illustrated in Figure 1.5. However, our objective is to predict a research topic or research topics of a given document and this research work can be extended to various application such as auto-assignment of research proposal to correct research topics for NSERC grant.

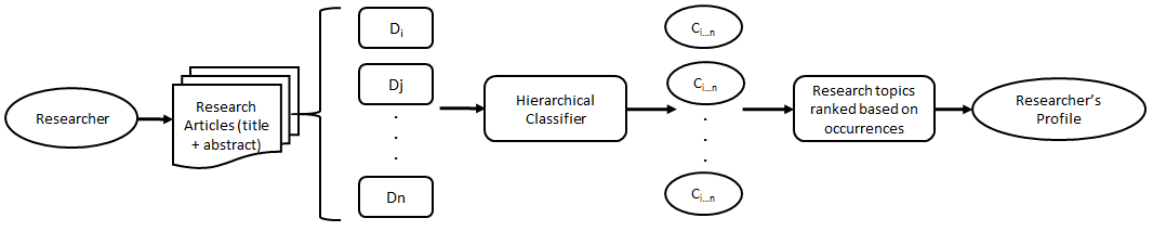


Figure 1.5: A general approach to test the methodology.

It is a fact that a research article can belong to multiple research topics. A research article can belong to research topic within the evaluation group or across the evaluation group and such articles can be multi-labeled. We have proposed a couple of methods to solve this problem which are discussed later in the Section 4.6.

We conducted various experiments with different classifiers such as Ridge classifier[2], Perceptron[3], Naive Bayes (Bernouilli and Multinomial)[4], and a 'linear' kernel based Support Vector Machine (Linear SVM[5]) using scipy[6] library on evaluation group 1501 (Genes, Cells, and Molecules), here onward referred as LSA. Of these, Linear SVM is scalable on sparse data and has better performance with different range of features compared to other classifiers, and therefore it is an ideal classifier for creating baseline and hierarchical models.

The performance of all the models used in the research was evaluated using different amount of training set and different evaluation measures such as precision, recall,

[2]http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html
[3]http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html
[4]http://scikit-learn.org/stable/modules/naive_bayes.html
[5]http://scikit-learn.org/stable/modules/svm.html
[6]https://www.scipy.org/

and F1-score. We created a baseline model consisting of research topics of NSERC evaluation groups as classes for flat classification. The performance of the flat classifier is compared with different hierarchical models. The output from the hierarchical classifiers based on different text representations is an input to the consensus methods which outputs a single class is compared with baseline classifier. The limitations and current issues with current approaches and methods used in the research are discussed.

The contributions of the research are as follows:

- Classifying a research article into a research topic using hierarchical classifiers based on pre-defined taxonomy.
- The use of features such as concepts and categories over BOW from Wikipedia to enrich document representation.
- A research article can belong to multiple research topics within or across evaluation groups and therefore, two methods are proposed to multi-label a research article.

This thesis discusses related work on finding an expert (Section 2.1), disambiguation of term(s) using Wikipedia (Section 2.2), and hierarchical classifiers (Section 2.3) in Chapter 2. In Chapter 3, different stages of the proposed methodology are schematically represented. The stages of the methodology such as document representations (Section 3.1), hierarchical classifiers (Section 3.2), predicting the class (Section 3.3), and consensus methods (Section 3.4) are discussed. The data set size, results from predicting the class approach, performance of different hierarchical models, baseline model, and consensus methods are shown and discussed in Chapter 4. In this Chapter, we have investigated the result of poor performance on use of concepts and categories by visualizing the data on the 2-Dimensional (2D) plot using t-Distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008). The Section 4.6 on multi-labeling a document includes various insights for the scope to multi-label a research article and two methods are proposed to multi-label a research article. In the last Chapter 5 of the thesis, highlights the limitations of current work, and directions for the future work.

Chapter 2

Related Work

There has been much work on finding an expert and research on this topic is now fairly known. In this chapter of the thesis, we will highlight previous work on finding an expert. Since our work involves classifying documents/research articles into one of the research topics of the evaluation groups defined by NSERC, we will cover previous work on hierarchical classification. In most of the natural language processing applications disambiguation of the text is a known problem and much research is done for disambiguation of mentions which consists of one or more terms from the text. One of the famous disambiguation approaches involves the use of Wikipedia concepts, also know as Disambiguation to Wikipedia (Mihalcea and Csomai, 2007) is discussed in this chapter.

2.1 Finding an expert

Finding an expert, the person who has the knowledge in one or more domains, is the task of profiling of an expert and responding to the user's query. Finding an expert involves creating a profile of each user first and then searching the expert based on user's query. Another approach is to find an expert based on user's query and matching expert's documents such as blogs, articles and question-answering repositories. These problems are more related to information retrieval tasks.

Many researchers have used external sources to train the model and then predict the researchers' expertise. (Chen et al., 2013) has used CitSeer[1] library to build expert recommendation system for computer science. They used n-grams of a title of each article to create candidate key-phrases and expanded using Wikipedia hyperlinks. (Charlin et al., 2012) proposed a framework to assign paper-to-reviewers

[1]http://citeseerx.ist.psu.edu/index

using suitability score defined as a relevance measure for a pair of reviewer and paper. Using learning methods such as Language Model (LM), Linear Regression (LR) and Bayesian Probabilistic Matrix Factorization (BPMF) affinity score between paper and reviewer is estimated from partially scored samples. The assignment problem is then solved using Integer Programming (IP) approach. Their approach has shown improvement in the result on two datasets of conference papers. A similar application of finding expertise is done using DBLP[2] bibliography data and Google Scholar[3] by (Deng et al., 2008). There are various sources such as PubMed Central[4], AMiner[5], Citeulike[6], and Microsoft Academic API[7], which can be used for finding an expert. Of these datasets for expert finding, DLBP is used by (Moreira et al., 2013) to rank expertise and Aminer by (Tang et al., 2008) to retrieve profiles of researchers. Both of these applications are examples of information retrieval where for a given query, a list of researchers are returned.

Finding an expert is not limited to querying scholarly databases, but also to question answering repositories. In this expert retrieval task, a user asks a question and their answers are extracted using topic modeling approaches used in (Riahi et al., 2012) and (Yang et al., 2013). Answers are aggregated for each user and by applying topic modeling to create expert's profile. This is then used to list experts for a given question asked by a user.

Our work is about creating a profile of each expert based on evidences. This profile is the list of research topics. A profile of each expert is based on the output from the classifiers for all of their documents and therefore, it is a classification problem.

Our work involves the use of controlled vocabularies that defines each research topic. A very closely related work using controlled vocabulary from IEEE[8] on visualization is done by (Isenberg et al., 2014). They have used user-defined keywords,

[2]http://dblp.uni-trier.de/
[3]https://scholar.google.ca/
[4]https://www.ncbi.nlm.nih.gov/home/develop/api.shtml
[5]https://aminer.org
[6]http://www.citeulike.org
[7]http://academic.research.microsoft.com
[8]http://www.ieee.org

IEEE assigned keywords by group of professors, IEEE automated system (INSPEC) and user selected keywords from IEEE paper submission form (PCS) to create a visualization of keywords over a period to 10 years. They have used clustering approach on articles submitted to five different conferences to group them into IEEE defined keywords. The contribution is not limited to grouping the articles, but also enable visualizing and maneuvering of all keywords in IEEE keywords set that allows researchers to select more effective keywords. One of the objective of the research is to bring all the articles published in "Visualization" domain to a common vocabulary.

A research, (Beel et al., 2016), on the need to create a common framework for accepting research papers, a common terminology and system that enables exchange of information between researchers is evaluated by questioning the quality of research done in the past. It statistically highlights the short-comings in the research papers on research-paper recommender systems published in the past. The short-comings discussed by (Beel et al., 2016) ranges from selecting a data set, inappropriate methodologies and baseline models, evaluation parameters and variation in user study which affects the reproducibility, use of promising approaches and overall quality of the work. The future work of our research involves creating a recommender system for researchers to help them create profile. The literature survey on research paper recommender system will help to overcome weaknesses posed by the authors.

2.2 Disambiguation to Wikipedia

Traditionally, text classification is based on BOW and each document is represented using Term Frequencies (TF) and Inverse Document Frequency (IDF) as product of TF and IDF. Each term of the document in a vector space of TF.IDF is independent of another term in the same document. This technique has many problems: (i) a meaningful phrase or multi-word mention breaks into individual word and its meaning is lost, (ii) it ignores the position of the word and therefore ignores the semantic relatedness, (iii) it treats synonymous words as separate entities and polysemous words as one single component (Wang and Domeniconi, 2008). These limitations do affect the performance of the classifier and little can be done to improve by pre-processing.

n-gram words have been used to address some of the limitation but it is computationally expensive. Therefore, it is essential to incorporate semantic information and conceptual relatedness measures to be able to enhance the prediction capabilities of classification algorithms.

In order to address these limitations, there has been much research to use semantic relatedness between terms and it is classified into three categories, knowledge-based systems, statistical approaches, and hybrid approaches (Altınel et al., 2015). Knowledge-based systems extract semantic knowledge from external sources such as WordNet, Wikipedia and MeSH.(Tsatsaronis et al., 2010), (Nasir et al., 2011), (Jing et al., 2010), (Mavroeidis, 2005), (Budanitsky and Hirst, 2006), (Lipczak et al., 2014). (Tsatsaronis et al., 2010) and (Nasir et al., 2011) used WordNet-based semantic relatedness measure of a pair of words called "Omioits" to create weighted TF.IDF vectors and incorporated into the semantic kernel. (Budanitsky and Hirst, 2006) made an extensive effort to measure and compare semantic relatedness and semantic distance of different approaches proposed for use in applications in natural language processing and information retrieval. However, the use of WordNet has shown good performance on some data sets, but it is restricted because it is manually built. Therefore, researchers started looking for another external source, such as Wikipedia.

Wikipedia is the largest and most visited encyclopedia in existence. The articles are densely linked to each other and with millions of incoming and outgoing links to Wikipedia articles. In our research, we are using Wikipedia links to an article for disambiguation to extract semantic knowledge of the terms/mentions. These are called concepts. Wikification is the task of identifying concepts and entities in the text by exploiting statistics behind in-links and out-links to Wikipedia articles (Milne and Witten, 2008).

Concept-based knowledge from an external source such as Wikipedia has been used extensively in the past to improve performance over BOW in Information Retrieval, clustering, and categorization tasks.

(Banerjee et al., 2007) used concept based representation to cluster popular news and blog feeds instead of overloading users with information. This concept-based representation has shown to improve performance over BOW representation. Similar use of Wikipedia concepts for clustering problem has been done by (Hu et al., 2008), (Hu et al., 2009), (Huang et al., 2009), and (Huang et al., 2009). Recent work using WordNet is done by (Altlnel et al., 2013), (Altinel et al., 2014), and (Poyraz et al., 2014) where they have created higher-order semantic kernel for text classification.

Wikipedia-based concepts were used by (Gabrilovich and Markovitch, 2006) to show that these vector representations can improve text classification results over BOW and in later research, (Gabrilovich and Markovitch, 2007) proposed a new approach, Explicit Semantic Analysis, by extending Latent Semantic Analysis using Wikipedia-based concepts to measure semantic relatedness between fragments or long text of the natural language. (Wang and Domeniconi, 2008) used Wikipedia-concepts to built the semantic kernel for text classification. Their results show improved performance over BOW representation using Wikipedia enriched representation and were further improved using Wikipedia-based semantic kernels. Wikipedia concepts can be used as an auxiliary classifier based on concepts with BOW concepts. (Yun et al., 2012) created two-layer text classification framework based on syntactic and semantic representation of Vector Space Model (VSM) and outputs from these classifiers are combined to finally predict the class of each test samples. Term VSM and concept VSM of training samples for each class are averaged to compute the centroid for each class and then cosine similarity between centroid and test samples is measured which results in k-dimension vector representation for each document, called the compressed representation. Predicted class at the first level for each document and corresponding compressed vector are aggregated as test samples and top level classifier predicts the final class.

2.3 Hierarchical Classification

Unlike flat classification approach, hierarchical classification considers parent-child class relationships which discriminate classes at each level and progressively moving

down the hierarchy. These types of classification discriminate among a large number of classes from different parents. A hierarchical classifier uses a pre-defined taxonomy as discussed in (Silla and Freitas, 2010) and originally defined in (Wu et al., 2005) as a binary relation over the set of finite classes C and relation being identified using "is-a" relationship. (Wu et al., 2005) defined "is-a" relationship as both transitive and anti-reflexive, and (Silla and Freitas, 2010) added an asymmetric relationship to it.

The mathematical representation of the properties mentioned are as follow:

- Tree with one element, i.e root is the greatest element.
- For all $c_i, c_k, c_j \in C,\ c_i \Rightarrow c_k\ and\ c_k \Rightarrow c_j,\ then\ c_i \Rightarrow c_j$ (transitive).
- For all $c_i \in C,\ c_i \not\Rightarrow c_i$ (anti-reflexive).
- For all $c_i, c_j \in C,\ if\ c_i \Rightarrow c_j\ then\ c_j \not\Rightarrow c_i$ (asymmetric).

A pre-defined taxonomy/class structure is a valid structure if all these properties are satisfied. The classification where the intermediary classes are created on the fly is not a valid taxonomy (Silla and Freitas, 2010). We have used pre-defined taxonomy of evaluation groups and research topics from NSERC that can be further divided to create a denser tree structure. Current structure defined by NSERC is two level and it satisfies all relationship properties.

In the real world, not all classification problems can be addressed using flat classification. Many problems have a structure in the form of hierarchy/tree and sub-trees which may have different height as shown in Figure 2.1. To add further complexity, node in the hierarchy can be related to another node within the sub-tree or across to create Directed Acyclic Graph (DAG). Classification in the hierarchy is not necessary to predict leaf node and hierarchical classification can be up-to mandatory leaf-node prediction or non-mandatory leaf-node prediction. However, in our problem, we did not have such complexity and have a tree with the maximum height of length 2.

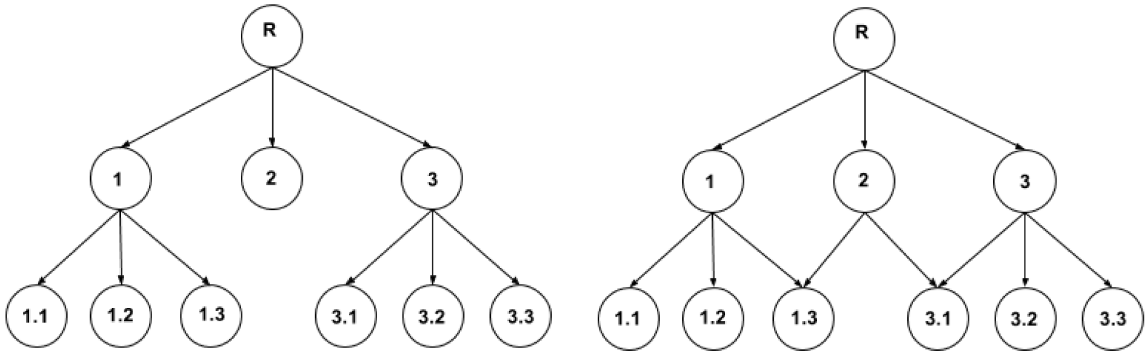


Figure 2.1: Example of simple tree (left) and DAG tree (right) based hierarchical class structure.

In our hierarchical structure, we are interested in predicting leaf node class which represents research topics of the evaluation groups defined by NSERC. Besides the structure of hierarchical model, based on type of local information used, it is categorized into the local and global classifier. Local classifier approach has three standard ways of using local information and they are Local Classifier per Node (LCN), Local Classifier per Parent Node (LCPN), and Local Classifier per Level (LCL) (Silla and Freitas, 2010).

A local classifier per node, LCN, uses local information where each node is trained as a binary classifier of positive and negative samples. This type of approach can use any training methods defined in (Silla and Freitas, 2010). Early work on hierarchical classification using LCN is done by (D'Alessio et al., 2000) to improve speed and F-measure. Web, a heterogeneous collection of web content, has hierarchical structure and (Dumais and Chen, 2000) used LCN for classifying text. They have used multiplicative decisive rule and Boolean decision rule based on some threshold at the top-level to combine the result and have improved result over flat classification. In all these classification approaches, classifiers can predict mandatory leaf node or non-mandatory leaf node or both as in (Sun and Lim, 2001). In threshold based top-down approach, the classifier may incorrectly classify a document into incorrect class at the top level or fail to meet threshold at the top level. In either of the case, it suffers from blocking problem. (Sun et al., 2003), and (Sun et al., 2004) proposed

performance measures and blocking measures to assess the contributions of misclassified documents and methods to mitigate blocking problem, respectively. There are among others who have worked on hierarchical classification, LCN, are (Liu et al., 2005), (Wu et al., 2005), (Cesa-Bianchi et al., 2006), (Cesa-Bianchi et al., 2006), and (Esuli et al., 2008). This type of hierarchical classification is one of the most widely researched. Also, there has been many works on Directed Acyclic Graph (DAG) by various researchers such as (Jin et al., 2008), and (Guan et al., 2008).

Local classifier per parent node LCPN, a type of using local information, is build by training parents at the same level or same level and descendants, except leaf nodes. This type of local approach is tested from a top-down, but it is not mandatory to follow this approach. Most suitable ways of training this type local classifier is "siblings" and "exclusive siblings" policy (Silla and Freitas, 2010). An extension of this type of approach, called "selective classifier", is proposed by (Secker et al., 2007). In their approach, they call it a "select top-down approach", but renamed as "select classifier" by (Silla and Freitas, 2010) because it selects classifier at each parent class nodes with highest classification accuracy. (Holden and Freitas, 2008) proposed an optimized algorithm using swarm intelligence to select classifier by doing a global search that considers entire tree structure at once. Improvements over selective classifier approach is also done by (Silla and Freitas, 2009) and (Secker et al., 2010). Most recent work on LCPN by (Ramírez-Corona et al., 2016) predicts non-mandatory leaf-node by considering all possible paths from top-down and pruning path based on minimum probability threshold. Similar work is done by (Hernández et al., 2014) where they have used Information Gain to prune the path to predict non-mandatory leaf node. Our local hierarchical classification approach uses this type of local information and exclusive sibling policy to train the model and classifies a document from top-down into mandatory leaf node class.

Local classifier per level, LCL, uses local information by training one multiclass classifier for each level independent of the parent nodes. This is the least used type of local information. (Cerri et al., 2014) worked on multi-label hierarchical structure using Hierarchical Multi-label Classification with Local Multi-Layer Perceptron

(HMC-LMLP), previously proposed in their work (Cerri and de Carvalho, 2011) and (Cerri et al., 2011), and improved result from previous work by altering parameter values. HMC-LMLP is a local HMC method where it makes predictions at each level and output from the previous level is an input for the Multi-Layer Perceptron network associated with the next level. In their previous work, they suggested two alternative, the Back-propagation algorithm and the Resilient back-propagation algorithm. In their very recent work (Cerri et al., 2016), they proposed new hierarchical multi-label classification method using multiple neural networks for classifying protein function. Similar work for multi-label categorization is done by (Madjarov et al., 2016) using Support Vector Machine and Random Forest.

Global approach overcomes drawbacks of a local classifier that it suffers from blocking problem (Silla and Freitas, 2010) where due to threshold used at higher level in local classifier, the classification may stop at intermediate level without reaching the leaf node. Global classifiers trains all the nodes in the tree simultaneously to have one classification model and this is relatively complex. Each test sample is simultaneously applied to each node in the hierarchy and thereby eliminating the blocking problem. There is limited research on hierarchical classification that uses global information. One of the work by (Levatić et al., 2014) is on multi-label hierarchical structure using trees. (Borges et al., 2013) proposed a new algorithm, Competitive Neural Network (HC-CNN), and compared its performance on Global-Model Naive Bayes on eight protein function dataset. Similar work on protein function prediction is done by (Alves et al., 2008), where they have proposed a new algorithm, Multi-label Hierarchical Classification with an Artificial Immune System, that allows multi-label identification and hierarchical classification. It has two versions of algorithms, one builds a global classifier that predicts all classes while other builds local classifier to predict each class.

Many work has been done in the past to deal with hierarchical structure. In recent years, hierarchies have become very popular for organizing text documents such as web content, and Wikipedia. These hierarchical structures have as large as hundred thousand categories and millions of documents. The challenge posed by such

complex hierarchical structure is not just the sparsity, but the problem arise dealing with imbalance in data across classes at different levels, complexity to train, and complex relationships between categories. To deal with this problem there has been various competitions such as BioASQ[9] challenge on large-scale biomedical semantic indexing and question answering, and Large Scale Hierarchical Text Classification (LSHTC[10]) challenge series which aims to assess and solve hierarchical problem by involving larger research community. In recent challenge competition LSHTC-4, the winning team, (Puurula et al., 2014), used ensemble of sparse generative models extending Multinominal Naive Bayes. It performs classification by predicting instances per label. A trained regression models on different classifiers are used to approximate optimal weights per label in the data set. The Linear Regression uses variants of Feature-Weighted Linear Stacking by distributing the weight of 1 uniformly to different baseline classifiers with maximum score.

[9]http://bioasq.org/
[10]https://www.kaggle.com/c/lshtc

Chapter 3

Methodology

In this chapter, we discuss the methodology. Figure 3.1 is used to classify a research article into one of the research topics defined by NSERC. Research journals are used as the source of titles and abstracts of the articles for each research topic. NSERC defines twelve evaluation groups and each evaluation group has research topics with keywords. Using this pre-defined taxonomy, a hierarchical classifier is created. We have used two different classifiers based on Wikipedia concepts and categories to improve the performance of hierarchical classifier on BOW. The output from these three classifiers is combined using different approaches to finally predict up-to three research topics for each document.

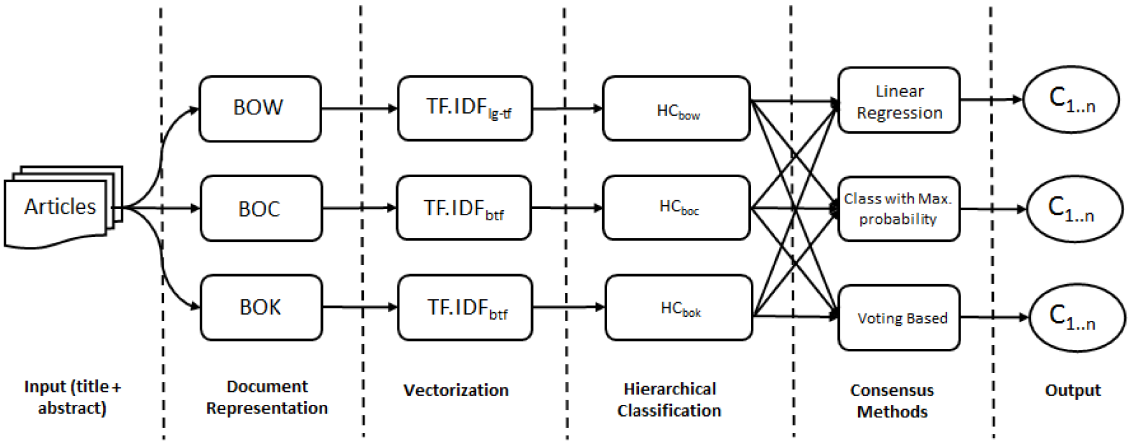


Figure 3.1: A methodology used to train and classify research article into a research topic. Each article is a combined text of the title and an abstract as an input to the methodology which is converted to different document representation and vetorized. These document vectors are an input to hierarchical classifiers and their outputs are generalized to output a single class using consensus method. The training process of the methodology is similar expect the output stage. $lg-tf$ stands for log normalized term frequency, btf means binary term frequency, $C_{1..n}$ is any of the n classes.

In the following sections, a description of each step in the methodology is explained

starting with document representation, hierarchical classifier, approaches to combine different outputs from the hierarchical classifiers, and multi-labeling a document.

3.1 Document Representation

Textual corpus is generally represented as BOW for text analysis, which is then converted into term frequency (TF) and inverse document frequency (IDF) (Manning et al., 2008). Each document is converted into a vector of terms calculated using a product of TF and IDF. To improve our representation of the documents, a semantic knowledge, called concepts, from Wikipedia is extracted using Wikipedia Miner Toolkit (Milne and Witten, 2013). Additionally, Wikipedia categories using Sunflower, an extended version of Tulip, are also retrieved (Lipczak et al., 2014).

3.1.1 Bag of Words

The traditional approach involves the use of Bag-of-Words BOW for text analysis which is a statistical measure of terms. Each document's title and abstract is preprocessed to remove English stop words[1], duplicate records/articles, articles with no abstract, and the text is stemmed using Potter Stemming. Then, for each document d a weight for the term t is assigned by calculating the number of occurrences of t in d. This weighting scheme is called term frequency ($TF_{t,d}$). Since there is no restriction on length of abstract, there can exist high frequency of term which may dominate during classification. To diminish such effect, modified weighting scheme called log normalization ($1 + TF_{t,d}$) is used. Then, an inverse document frequency ($IDF_{t,d}$) is calculated for each term to understand how common or rare is the word in the corpus of all the documents D as shown in Equation 3.1. In Figure 3.1 the $TF.IDF_{lg-tf}$ in vectorization stage represents the product of log normalized TF and IDF. It is calculated as logarithmic inverse of frequency of term t in a document d over the corpus D as shown in Equation 3.1.

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

[1] www.nltk.org

Finally, a product of TF and IDF is calculated for each $t \in D$ as shown in Equation 3.2

$$TF.IDF_{t,d,D} = TF_{t,d} * IDF_{t,D} \tag{3.2}$$

An important advantage of doing product is that the term with high frequency and low document frequency will have a high score and eliminates common terms by assigning a lower value (Manning et al., 2008).

This type of document representation has limitations. First it assumes terms independent from each other and breaks the meaning of the terms that appears together. It ignores underlying semantic and syntactic connection. It means the order of the terms in the text is ignored by this representation. It treats polysemous word(s) as single entity and synonyms as different entity (Wang and Domeniconi, 2008).

3.1.2 Bag of Concepts

To supersede the drawbacks of BOW, we have used Wikipedia knowledge base for disambiguation of term(s). Wikipedia is the largest publicly created network of in-links and outlinks of articles which removes disambiguation in the text by referring a term(s) to the right article. We call these terms as concepts.

Using Wikipedia Miner Toolkit (Milne and Witten, 2013), concepts for a given document is extracted and new vector representation for each document using TF.IDF is created. In this representation, for each document, a Boolean term frequency is calculated. In the process of concept identification, for each occurrence of the concept in the text, a single instance is retrieved. Each of these concepts identified has unique identification number and name of the concept which links to the Wikipedia article. A score is assigned to each concept based on similarity to the text and this score is the probability. Wikipedia is good source for disambiguation, but it also have irrelevant concepts. The list of concepts for each wikified text needs to be pruned by selecting appropriate probability threshold.

Machine learning is the subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959). Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is unfeasible; example applications include spam filtering, optical character recognition (OCR), search engines and computer vision.

Figure 3.2: Sample input of text for wikification to Wikipedia Miner Toolkit.

Machine learning is the subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959). Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is unfeasible; example applications include spam filtering, optical character recognition (OCR), search engines and computer vision.

Figure 3.3: Wikified text from Wikipedia Miner Toolkit.

In the Example in Figure 3.2, a human reader understand what terms goes together and can disambiguate the meaning of the term(s) in the text. However, machines are not capable to disambiguate such information and therefore external knowledge such as Wikipedia is used that maps term(s) to a concept as shown in Figure 3.3. It not only identifies the terms but also disambiguate them to correct interpretations. Each concept has a unique identity, probability score, and the name of the concept (title of Wikipedia article) in the output from Wikipedia Miner Toolkit. The name of the concept may consists of more than one term and therefore each name of the concept are joined using "_".

Bag-of-Concepts (BOC) is created for each document using the name of the concepts. Concepts for each document are represented as TF.IDF vector calculated as discussed in Sub-section 3.1.1, except for the Term Frequency (TF) where a binary weighting scheme is applied. In this weighting scheme each concept is assigned a score of either [0,1] based on its presence in the document.

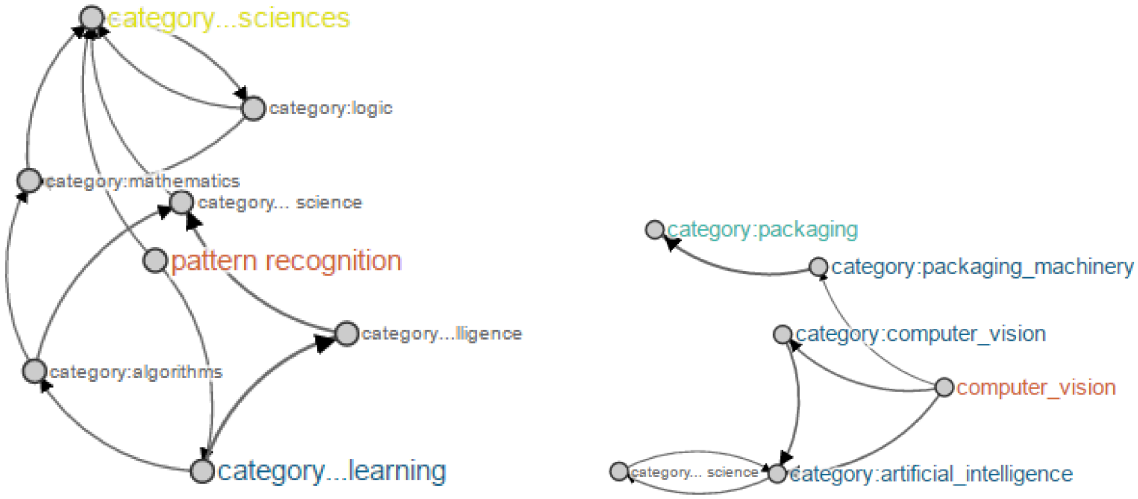
3.1.3 Bag of Categories

Wikipedia is a densely linked network of information manually built over years. The information is not limited to concepts and link to articles like typical web structure, it also categorizes the articles. The concepts identified by Wikipedia are not 100% accurate. Some amount of noise is expected and this may affect chances to improve classification. Therefore, another representation is chosen to enrich BOW further. In the Sub-section 3.1.2, we mentioned that concepts are linked to Wikipedia articles. Each Wikipedia article has one or more categories representing the width and categories are linked to other categories representing the depth. We have extracted categories for each concept in the text to enrich BOW representation.

Wikipedia based tool, Sunflower, an extended version of Tulip (Lipczak et al., 2014), extracts categories for corresponding concepts. Tulip uses many languages to decipher the correct categories of the concept. To each category a score is assigned by Sunflower based on relatedness with the concept. Based on the given depth and width, it retrieves the categories and can be visualized as a graph as in Figure 3.4a. For each concept in the text, a set of categories is retrieved. All these categories from all the concepts for each document are combined to create a BOK. BOK are represented as vectors of TF.IDF where binary representation scheme is used for TF as calculated in Sub-section 3.1.2.

A concept is an input to Sunflower to which it outputs a list of categories each with a value of relatedness. An example in Figure 3.4a, the concept "Pattern Recognition" is directly linked to categories, "Sciences" (center-trimmed category at the top), "Machine Learning" (center-trimmed category at the bottom) and further to other categories. The problem with using categories is the level of depth and width. The depth of the tree is the distance in number of levels. Whereas, width is the direct relationship with the concept. For an example, in Figure 3.4b, the concept "computer vision" have categories "packaging" and "packaging_machinery" which is an application of computer vision in the packaging industry.

It is understood from these examples that there could be categories which are not



(a) Concept: Pattern Recognition (b) Concept: Computer Vision

Figure 3.4: Categories from Sunflower for a given concept.

relevant or too abstract. It is very important to do vertical and horizontal pruning of the tree to retrieve the relevant categories. Sunflower retrieves the list of categories with relatedness score. The depth and width of categories can be controlled using Sunflower tool that enables pruning based on parameters.

3.2 Hierarchical Classification

Local classifiers are grouped based on how they use local information. Local information is used in three ways in local hierarchical classification: a local classifier per node (LCN), a local classifier per parent node (LCPN), and a local classifier per level (LCL) (Silla and Freitas, 2010). For our research, we have used LCPN for hierarchical classification and training is performed by leveraging the “exclusive siblings” policy. A local classifier is trained as a binary classifier consisting of node to be trained as a positive and rest as negative samples where the positive and negative samples are from the siblings and descendants of the same parent. This is called ”siblings” policy. The more restrictive form, ”exclusive siblings” policy, of training classifier would be to not allow any descendant nodes in the siblings policy to be a part of positive and negative nodes or samples.

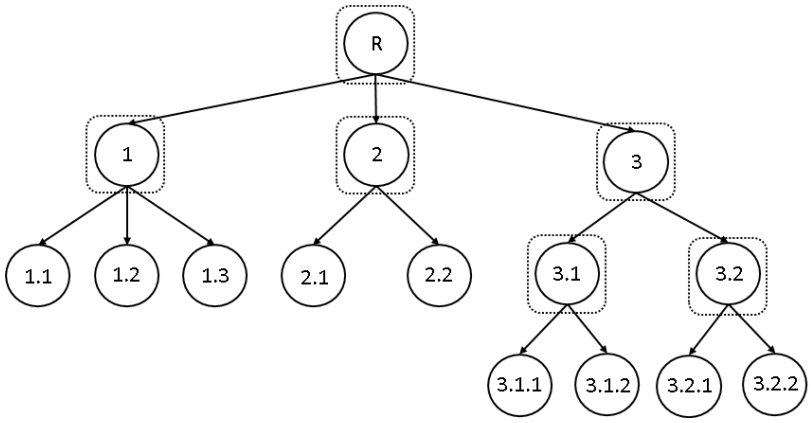


Figure 3.5: An example of local classifier per parent node LCPN.

To illustrate an "exclusive siblings" policy of training a local hierarchical classifier, refer to Figure 3.5. Suppose node 2 is to be trained according to LCPN "exclusive siblings" policy, then $N_{(2)}$ will have positive samples ($N^{+}_{(2)}$) from node $\{2\}$ and negative samples ($N^{-}_{(2)}$) from nodes $\{1, 3\}$. But, in the case of "siblings" policy, $N_{(2)}$ will have $N^{+}_{(2)} = \{2,\ 2.1,\ 2.2\}$ and negative samples $N^{-}_{(2)} = \{1,\ 1.1,\ 1.2,\ 1.3,\ 3,\ 3.1,\ 3.2,\ 3.1.1,\ 3.1.2,\ 3.2.1,\ 3.2.2\}$.

Not all policies for training local classifier are suitable for different types of local classifiers. LCN can use any of the mentioned policies, LCPN can use either "exclusive siblings" or "siblings" policy, and "exclusive siblings" policy is suitable for LCL. A One-vs-Rest (O-v-R) classifier by default uses "exclusive siblings" policy for training the data. For our research and for hierarchical classification, we have used O-v-R classifier for training each parent node. By default, O-v-R classifier uses "exclusive siblings" policy and a O-v-R classifier is created for top-level (Evaluation Groups), and three at leaf-level for all research topics of each evaluation groups.

The advantage of hierarchical classifier is its better performance over flat classifier. The hierarchical classifier breakdowns the problem into sub-problem by making a decision at the prior level before it moves down to the child nodes. This results in pruning the tree vertically and less number of nodes are involved in classification as it traverses down the hierarchy. In contrast, a flat classifier trains all the nodes at

the leaf-level and performance is affected when there are a large number of classes to train. The poor performance also accounts for imbalance in training samples for positive and negative classes such as in O-v-R classifier. The NSERC hierarchical classification problem can also be solved using a flat classifier, but the performance is expected to degrade when a number of classes are added to the current implementation. Since the depth of the tree is limited to two-level and large width of the tree, the hierarchical classifier will perform better by pruning tree at the top-level and making a decision on the sub-tree it traverses.

3.3 Predicting The Class

This section of the thesis discusses different approaches used to predict a mandatory leaf-node. We have discussed two methods to predict the class at the leaf-level. One of the approaches is referred as path with maximum probability at each level of a parent node and the other as path with maximum product of the probabilities.

In hierarchical classification, LCPN, we use O-v-R classifier for each node that outputs a confidence score for each class to which a document belongs. These confidence scores are probabilities from O-v-R classifier. In the first approach, We follow the path with maximum probability at each level of a parent node and further predict probabilities of the child nodes. A class at the leaf node with maximum probability is the predicted class. This method is applied to all hierarchical classifiers based on document representation. The approach to predict a class based on this process is illustrated in Figure 3.6 where a hierarchical classifier predicts the node LSA of the parent node R with maximum probability and moves further down the hierarchy and predicts the node with value 0.40.

In the second approach, a parent-child nodes in the hierarchical structure are combined by taking the product of the probabilities for each possible path from the root to the leaf node and predict the class with maximum probability. This method considers a parent-child relationship along the path and this approach has been used

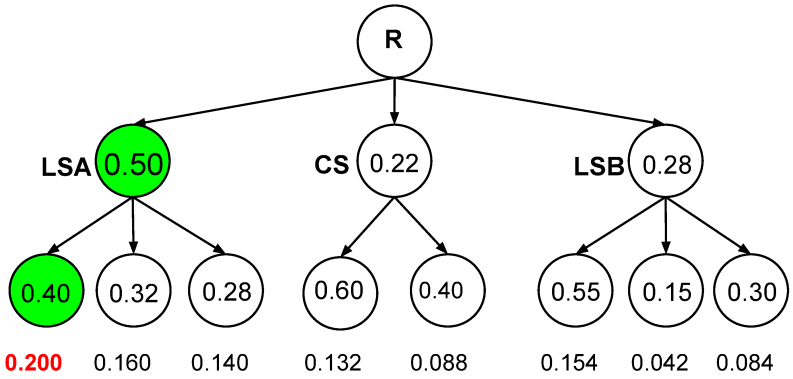


Figure 3.6: The nodes with green color is the path with maximum probability at each level for a parent node. The number at the bottom below the leaf-nodes are the product of the probabilities for each path from top-down and the node above red color value is a class with maximum product of the probabilities.

by (Hernández et al., 2014). This approach can be defined as given a set of classes $C = \{c_1, c_2, ..., c_l\}$ of path p where l is the leaf node starting from root node 1, the product of each path p for a vector v using a chain rule is as follow:

$$P(c_1, c_2, ...c_l|v) = P(c_1|c_2, c_3, ..., c_l, v)P(c_2|c_3, c_4, ..., c_l, v) : P(c_1|v) \tag{3.3}$$

In a given taxonomy, the classes are independent of each other and classes at lower level is a subset of higher level node. Equation 3.3, can be simplified as follows:

$$P(c_1, c_2, ..., c_l|v) = P(c_1|v)P(c_2|v)...P(c_l|v) \tag{3.4}$$

The Equation in 3.4 calculates the product of the probabilities for each path and the path with maximum probability is selected and the class at leaf-node as the predicted class as in Figure 3.6. The values below each leaf-node is the product of the probabilities along the path and the value in red color is the path with maximum of the product of the probabilities.

The main reason behind using maximum product of the probabilities is to address the drawback of local classifiers which allows incorrect prediction at top level to propagate down the hierarchy to leaf-nodes. This product of the probabilities for each possible path from top-down is considered and decision is made by taking the path with maximum value. The problem and possible solution to solve such problem

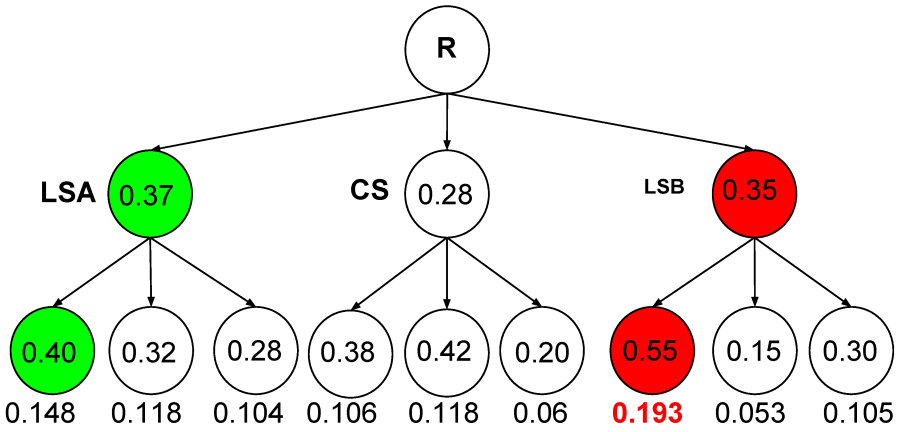


Figure 3.7: The nodes colored green is the path with maximum probability at each level of a parent node and in red is the path with maximum of the product of the probabilities.

is illustrated in Figure 3.7 where the path in red is selected and the node at the leaf as the predicted class.

3.4 Consensus Methods

The objective of extracting concepts and categories to enrich document representation based on BOW is to improve the performance of hierarchical classifier based on BOW. For each document representation, a hierarchical classifier is created. The output from each of these hierarchical classifiers is combined using different approaches based on some agreement.

The first method combines the predicted class from different hierarchical classifiers for different document representations to output a class with maximum probability as shown in Figure 3.8 and mathematically in Equation 3.5. HC in the Equation is a hierarchical classifier and $P(HC_{bow})$ is the probability of predicted class by HC_{bow}. $C_{i,bow}$ is the class with maximum probability predicted by *bow* from the set of classes C and i is the range from 1 to n where n is $|C|$.

$$C_{i..n} = argmax[C_{i,bow}, C_{j,boc}, C_{k,bok}] \quad where, \tag{3.5}$$

$$bow = P(HC_{bow}),\ boc = P(HC_{boc}),\ bok = P(HC_{bok})$$

$$1 \leq \{i, j, k\} \geq n \quad where,\ n\ is\ the\ number\ of\ classes$$

$$C_{i,bow}, C_{j,boc}, C_{k,bok}\ is\ based\ on\ max - max\ probabilities$$

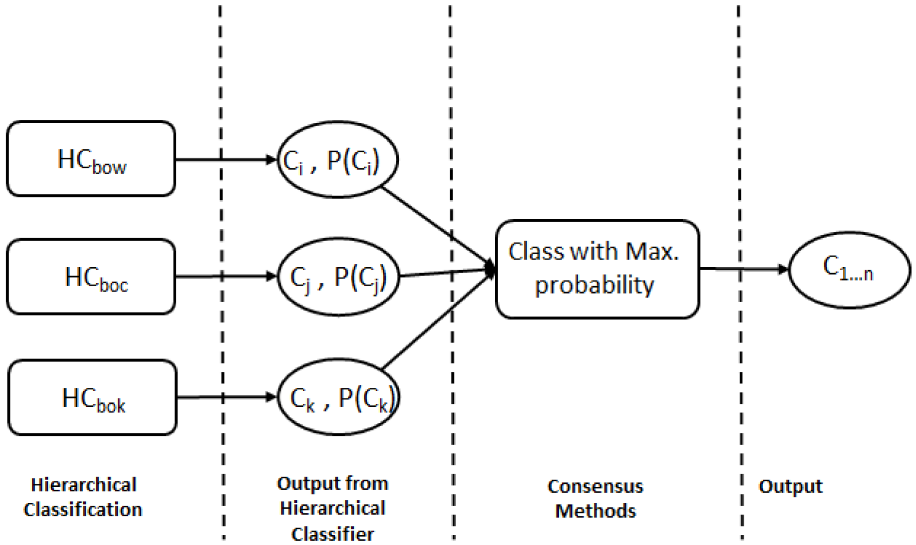


Figure 3.8: Partial methodology from hierarchical classification and consensus method to select a class with maximum probability predicted by hierarchical classifiers.

The second method combines predicted class from each type of a hierarchical classifier for different document representations to output a class with maximum votes. The tie cases in this approach are currently not addressed in our work and either of the class in tie is returned. The approach to predict a class based on votes is illustrated in Figure 3.9. $C_{i,bow}$ is the predicted class by a particular hierarchical classifier, HC_{bow} in this case, in Equation 3.6.

$$C_{i..n} = Max\_number\_votes[C_{i,bow}, C_{j,boc}, C_{k,bok}] \quad where, \tag{3.6}$$

$$bow = HC_{bow},\ boc = HC_{boc},\ bok = HC_{bok}$$

$$1 \leq \{i, j, k\} \geq n \quad where\ n\ is\ the\ number\ of\ classes$$

$$C_{i,bow}, C_{j,boc}, C_{k,bok}\ is\ based\ on\ max - max\ probabilities$$

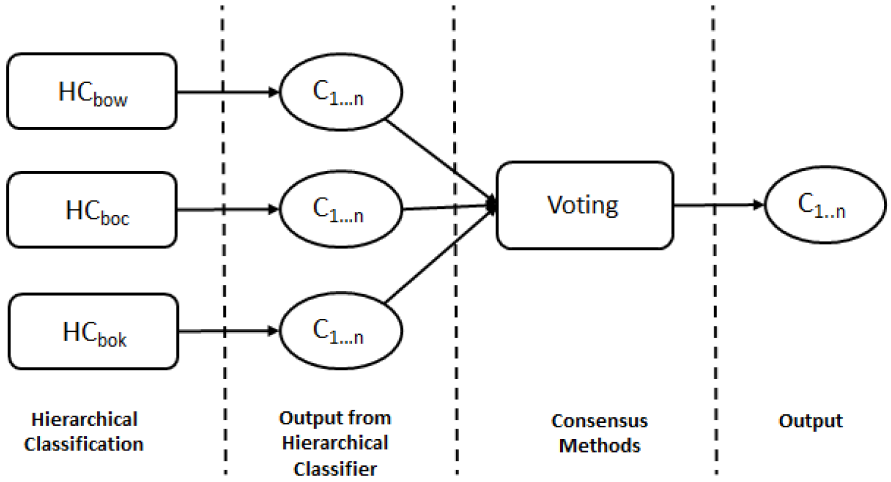


Figure 3.9: Partial methodology from hierarchical classification and consensus method to select a class with maximum number of votes.

In the third approach, the output from each type of hierarchical classifier for different document representations is considered for Linear Regression. The linear regressor is trained on the probabilities from hierarchical classifiers for different representations. The predicted classes from these hierarchical classifiers are discarded and original labels are used. Except the target, all other columns have continues values as probabilities and these target values need to be an ordinal values for Linear Regression. Using One-Hot encoding or label binarizer[2], each of these classes are converted to series of bit of length of number of classes and only one bit is 'On' at particular position and rest are 'Off'. The target class after encoding are represented as 31 column, the equivalent of number of research topics. An 'On' bit at a particular position corresponds to the label of research topic. Our objective of using this approach is to generalize the output from hierarchical classifiers on training data and test it using testing data. The training of Linear Regression is shown in Figure 3.10. The probabilities for each class from hierarchical classifiers are represented as $P(C_{x,i})$ where x is the index of the document in the data set and i is the index of the class in n classes such that $1 \leq i \geq n$.

The testing of a Linear Regression approach follows the same approach as other methods until hierarchical classification. Instead of the predicted class, a list of

[2]http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelBinarizer.html

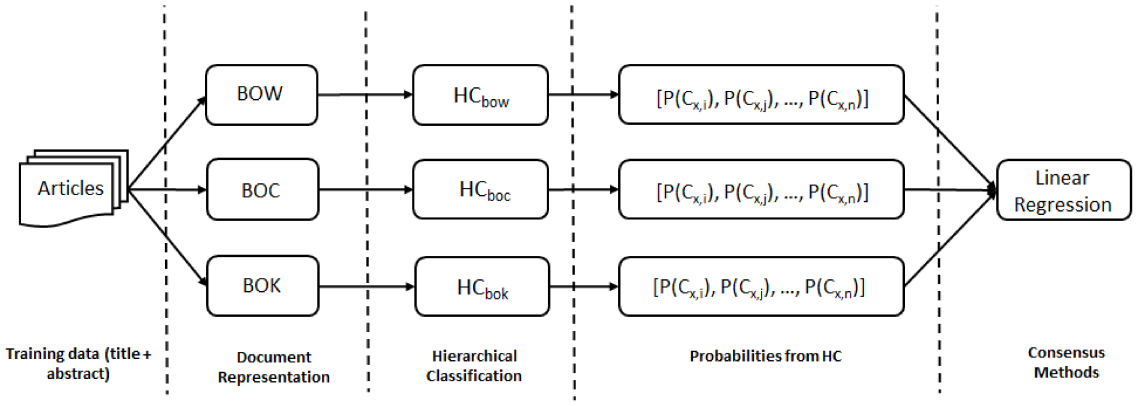


Figure 3.10: Training a Linear Regressor on probabilities predicted by a hierarchical classifier.

probabilities for each test sample from a hierarchical classifier for each document representation is averaged for each class and given as an input to trained linear regressor (refer Figure 3.11). Linear regressor predict one of the encoding and it is reversed to the name of the research topic.

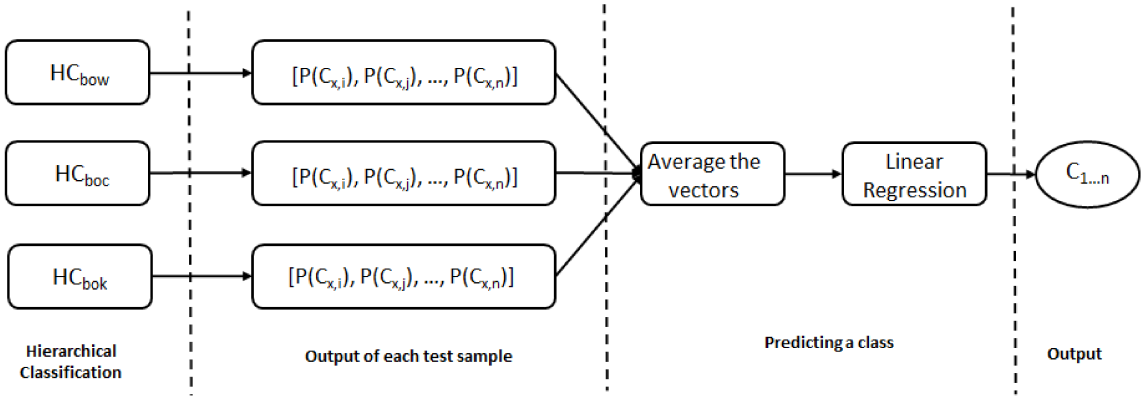


Figure 3.11: Partial methodology from hierarchical classification and Linear Regression to predict a class.

The output from each of these consensus methods is then compared based on evaluation measures such as precision, recall and F-measure. These outputs are also compared with baseline classifier and the hierarchical classifier based on BOW representation.

Chapter 4

Experiments and Results

4.1 Data Collection

In our research, we have used NSERC taxonomy as a pre-defined two-level hierarchical structure where the first level is the evaluation group and the second level is the research topic. For each research topic a set of journals are selected whose aims and scope matches the keywords of a research topic exclusively. For our research, we have used the title and an abstract of each research article of the selected journals.

An abstract of the research articles precisely describes the research. The title and an abstract of research articles are largely available in various online repositories such as Google Scholar, CiteSeer, Citeulike, Microsoft Academic API search, DBLP, ArnetMiner, and more. The real challenge is not in retrieving the data from these repositories, but finding a reliable and complete abstract, a labeled articles that conforms to common terminology across these repositories, and combining articles from different sources for all evaluation groups defined by NSERC.

It is challenging to merge the labeled articles from these free online databases, and the quality of the data will remain in question. To address this problem, a set of keywords defined by NSERC for each research topic (refer Table 4.1) are used to extract articles from journals and label articles with a research topic. It is practically impossible to label hundreds of thousand of research articles manually and therefore a reliable approach is used to address this problem. The problem is resolved by matching keywords in aims and scope, step 2 in Algorithm 1, of each journal K_{jour} with keywords defined for each research topic at NSERC K_{rt} such that $K_{jour} \subseteq K_{rt}$ and $K_{jour} \notin \{K_{rt_1}, K_{rt_2}, ..., K_{rt_n}\}$ where, a journal exclusively belongs to a research topic within an evaluation group. The high level of process of data extraction is illustrated in Figure 1.3.

Genes, Cells and Molecules			
Label	Research Topics	Keywords	Journals
LSA01	Immunology	Host-cell interactions; immune response; antigens; antibodies; host-pathogen interactions; immunogenetics; innate immunity; cytokines and antimicrobials; antigen presentation; inflammation; lymphocyte; neutrophil; monocyte; macrophage; sinus; thymus epithelium; lymph node; spleen; chemokine; interleukin; dendritic cell; B cell; T cell; plasma cell; mucosal immunity; immunoglobin; ecological immunology; Toll-like receptors; evolution of immune responses	Nature Reviews Immunology; European Journal of Immunology; Annual Review of Immunology; Advances in immunology; Trends in Immunology; Immunological reviews

Table 4.1: A research topic, label and keywords defined by NSERC and shortlisted journals.

To avoid the problems of merging articles from different online repositories and for the ease of retrieval of relevant articles for each research topic, a reliable source, Exlibris API[1] through Novanet, Inc[2], is used. For hierarchical classification, we selected three evaluation groups (1501 as LSA, 1502 as LSB, and 1507 as CS) and 31 research topics across these groups. The complete list of evaluation groups, research topics, and their keywords can be found at NSERC[3], and corresponding journals for each research topic in Appendix A.2. From 156 shortlisted journals, total 176,486 articles published in the year 2000 and later are retrieved. There are 7 research topics (LSA03, LSA10, CS02, CS10, CS13, CS16) across 3 evaluation groups that were deliberately skipped due to lack of journals or articles retrieved. This is the step 3 and 4 of the Algorithm 1.

Each article that is retrieved is labeled with a matching research topic label as

[1]https://developers.exlibrisgroup.com/primo/apis
[2]http://www.novanet.ca/
[3]http://www.nserc-crsng.gc.ca/Professors-Professeurs/Grants-Subs/DGPList-PSDListe_eng.asp

shown in step 5 of the Algorithm 1. In step 6, the title and the abstract of each article of the journal are then joined to create a text. In step 9, the text is wikified using Wikipedia Miner Toolkit(Milne and Witten, 2013) and concepts with probability ≥ 0.50 are mapped with an article and persisted in the database. Along with Wikification, in line 10, categories from Wikipedia are retrieved using Sunflower. The text, concepts and categories represents BOW, BOC, and BOK, respectively.

```
   Input  : A list of journals with ISSN
 1 for i ← 0 to journals do
 2     research_topic_id ← getMatchingResearchTopicId(journals[i].scope);
 3     articles ← getArticles(journals[i]);
 4     for j ← 0 to articles do
 5         if articles[j].year ≥ 2000 and articles[j].abstract ≠ ∅ and
            !ifExists(articles[j]) then
 6             articles[j].label ← research_topic_id;
 7             articles[j].text ← articles[j].title + articles[j].abstract;
 8             articles[j].stemmed ←removeStopWords(articles[j].text);
 9             articles[j].concepts ←wikifyArticle(articles[j].text);
10             articles[j].categories ← getCategories(articles[j].concepts);
11             persistInDb(articles [j]);
12         end
13     end
14 end
```

Algorithm 1: A function to retrieve data from each journal.

The function in the second line in the Algorithm 1 is manually accomplished. Each journal whose aims and scope matches the research topic are further verified from Dalhousie Libraries (via Novanet) and Microsoft Academic Search online interface that enlists the topics/subject described by each journal. More details on using these tools and verifying journals could be found in Appendix A.1. The enlisted keywords were

further checked for irrelevant keywords from other research topics. The enlisted topics/subjects were filtered result based on irrelevant keywords to understand whether it belongs to the research topic. After proper manual inspection, a journal is assigned to a research topic and articles in a journal are assigned. It is further pre-processed to remove duplicate articles in another language such as French, and articles without an abstract.

4.2 Evaluation Measures

We have used various classifiers such as O-v-R, One-Class, and hierarchical classifier, and evaluation measures, precision, recall, and F-measure for the performance of the model.

The precision is the number of true positives over sum of number of true positives and number of false positives as shown in Equation 4.1.

$$P = \frac{|T_p|}{|T_p| + |F_p|} \tag{4.1}$$

Recall is the number of true positives over sum of number of true positive and number of false negatives as shown in Equation 4.2

$$R = \frac{|T_p|}{|T_p| + |F_n|} \tag{4.2}$$

F-measure or F_1 is a single value representation for precision and recall, and it is a harmonic mean of precision and recall. The formula to calculate F_1 is show in Equation 4.3:

$$F_1 = 2\frac{P \times R}{P + R} \tag{4.3}$$

Accuracy is the number of correctly identified samples from the entire dataset and it is calculate as sum of number of true positives and true negatives over total number of samples $|D|$ as in Equation 4.4.

$$A = \frac{|T_p| + |F_p|}{|D|} \tag{4.4}$$

All the reported values are macro and weighted average. Macro-average is the average of values of the system on different sets and it is shown in Equation 4.5.

Whereas in weighted average we consider imbalance in the number of samples of different sets and each set is assigned a weight of the system of different sets are to be averaged are given weights for as shown in Equation 4.6.

$$Macro\_average = \frac{\sum_{i=1}^{n} V_i}{|D_i, D_j, ..., D_n|} \quad where, \tag{4.5}$$

n is a total number of data sets, V_i *is the value of the data set* D_i*.*

$$Weighted\_average = \frac{\sum_{i=1}^{n} W_i.V_i}{|D_i, D_j, ..., D_n|} \quad where, \tag{4.6}$$

n is a total number of data sets, V_i *is the value and* W_i *is the weight of the data set* D_i*.*

4.3 Choosing A Classifier

We have used various O-v-R classifiers to choose a classifier with best performance. We used 10-fold cross validation on five O-v-R classifiers; Ridge Classifier, Perceptron, Multinomial Naive Bayes, Bernoulli Naive Bayes, and Linear SVC. Linear SVC is selected as a classifier to create our models and a baseline model due to better performance on variable number of features. The results are shown in Table 4.2, and appendix Table B.1 and B.2 are on BOW from an evaluation group LSA.

	Weighted Average Score			
Classifier	Accuracy	Precision	Recall	F1
Ridge Classifier	0.8961	0.8953	0.8961	0.8944
Perceptron	0.8902	0.8882	0.8902	0.8894
Multinomial Naive Bayes	0.8522	0.8515	0.8522	0.8500
Bernoulli Naive Bayes	0.8458	0.8532	0.8458	0.8477
Linear SVC	**0.9046**	**0.9038**	**0.9046**	**0.9037**

Table 4.2: O-v-R Classifiers performance on Evaluation Group: LSA

4.4 Hierarchical Classifier

A hierarchical local classifier, LCPN, is created using "exclusive sibling" policy. For a set of classes per parent a O-v-R classifier is created. For our hierarchical structure,

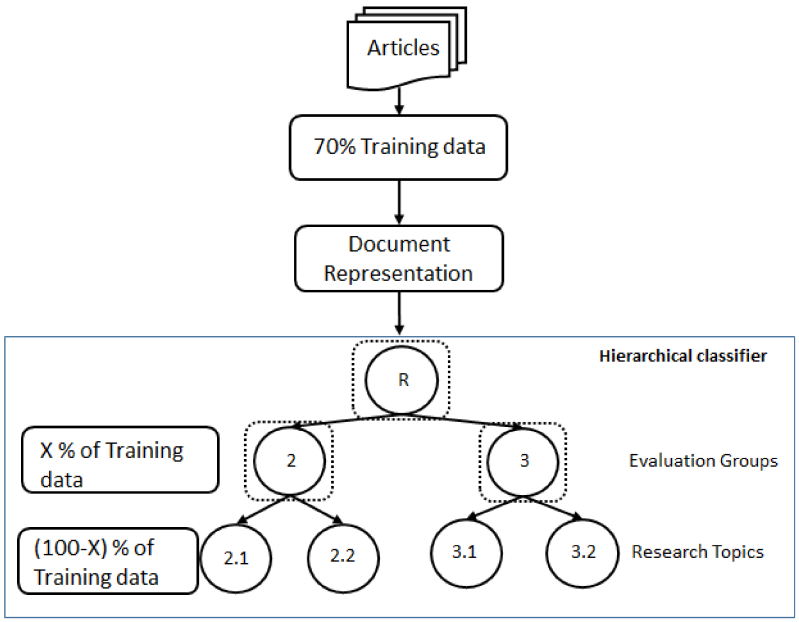


Figure 4.1: Schematic representation for training a hierarchical classifier.

at the top level, we have evaluation group and an O-v-R classifier is created by taking 10% of the data from each research topic data. The size of data set for the parent can be optimized for the performance. Then for each evaluation group, an O-v-R classifier is created using remaining data set. Entire hierarchical classifier is trained on 70% of the training data and performance of the model is evaluated on 10 random set of 70-30% train-test data. The process of training a hierarchical classifier is shown in Figure 4.1.

Hierarchical classifiers for the path with maximum probability at each level (refer Table 4.3) and path with maximum product of the probabilities (Table 4.4) for each document representation are compared. The results in Table 4.4 shows that the precision for path with product of the probabilities is higher for each document representation, but have poor recall and f-measure compared to hierarchical classifier based on the path with maximum probability at each level (Table 4.3). It has been observed that the prior approach suffers from an imbalance in the number of child nodes for different parents at the same level. For an example, in Figure 4.2, the node labeled green is the original label and the one with the red is predicted by product of the probabilities using the chain rule. Such problem occurs due to probabilities

Path with maximum probability at each level				
(Macro, Weighted)	Precision	Recall	F_1	Accuracy
BOW	0.7693, 0.8237	0.7523, 0.8214	0.7583, 0.8225	0.8174
BOC	0.6186, 0.6856	0.6034, 0.6812	0.6084, 0.6834	0.6785
BOK	0.6123, 0.6723	0.5976, 0.6745	0.6016, 0.6734	0.6717

Table 4.3: Path with maximum probability at each level

Path with maximum product of the probabilities				
(Macro, Weighted)	Precision	Recall	F_1	Accuracy
BOW	0.7782, 0.7923	0.6836, 0.7881	0.7092, 0.7702	0.7823
BOC	0.6414, 0.6614	0.5205, 0.6497	0.5444, 0.6245	0.6398
BOK	0.6465, 0.6600	0.5143, 0.6343	0.5353, 0.6184	0.6296

Table 4.4: Path with maximum product of the probability at each level

assigned to n classes of a parent node is normalized to the range of 0 to 1 and sum of their probabilities is 1. The path with maximum product of the probabilities may be predicted if the number of nodes are equal. The addition of one more node to parent node (LSA), for an example, could result is smaller value for node with value 0.65 than the current value, but higher than the others. Such cases are prevailing in our taxonomy and in current implementation which affects the performance of this approach.

In the same example, assume all the nodes for each parent are equal, values of the node in red and green are unaffected, and the original label is the node with value 0.65, then the exact inverse of the previous scenario is the drawback of the path with maximum probability at each level. Due to imbalance problem using hierarchical classifier based on product of the probabilities and poor performance compared to other approach, a hierarchical classifier based on path with maximum probability at each level of a parent node will be used from here onward for all comparison and result analysis.

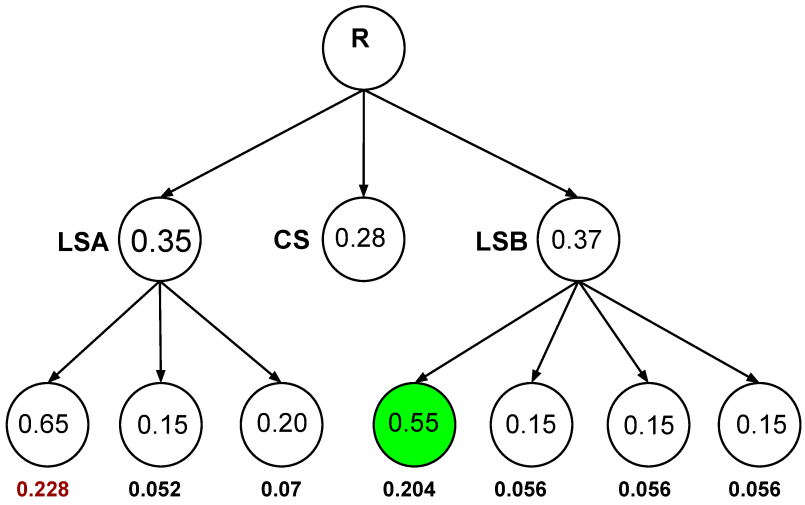


Figure 4.2: An example of imbalance problem for a parent at some level.

The hierarchical classification problem can be solved using flat classification, but flat classification will perform poor on large number of classes and when the number of samples for each class varies (Guo et al., 2008). The O-v-R classifier will be trained on severe imbalance in positive and negative sample for the nodes with less number of samples compared to other nodes. In contrast, hierarchical classifier takes advantage by pruning tree vertically and choosing less number of paths. Additionally, when the hierarchical tree is traversed downward the tree is pruned vertically because of the training policy used here. Each O-v-R classifier is trained for child nodes of each parent node and therefore has less number of classes to train and discriminate during test. Hierarchical local classifier does have drawback of being misclassified at higher level, but the problem could be resolved partially by taking threshold based approach at top level and choosing one or more paths. One can also choose to train the top level as good as possible. In our current approach, the top level makes an error of up-to 12% across different hierarchical classifiers based on different document representations of 10% of the 70% of the training data as shown in Table 4.5.

The drawback of flat classifier can be proved by simple test on different data sizes. In Table 4.6, the performance of flat and hierarchical classifier based on BOW is shown on different parent-child data size split. The hierarchical classifier is trained by taking x% training data for the parent node and (100-x)% training data for the

Macro and weighted averaged scores				
HC (top-level)	Accuracy	Precision	Recall	F_1
BOW	0.9512	0.9479, 0.9811	0.9424, 0.9512	0.9450, 0.9511
BOC	0.9028	0.8963, 0.9024	0.8864, 0.9028	0.8908, 0.9023
BOK	0.9006	0.8919, 0.9000	0.8845, 0.9000	0.8878, 0.9000

Table 4.5: The performance of hierarchical classifier at top level (Evaluation Groups).

leaf nodes. For the baseline model, the (100-x)% training data is used to train the model. It is observed that the performance of flat classifier drops more than hierarchical classifier when less amount of data and more number of classes are available for the classification task.

Macro averaged scores on BOW			
Parent-Child Data Split	Measures	Flat Classifier	HC
10% Parent - 90% Leaf-nodes	Precision	0.7826	0.7693
	Recall	0.7634	0.7524
	F1	0.7708	0.7584
50% Parent - 50% Leaf-nodes	Precision	0.7754	0.7673
	Recall	0.7469	0.7458
	F1	0.7546	0.7530
75% Parent - 25% Leaf-nodes	Precision	0.7588	0.7588
	Recall	0.7265	0.7328
	F1	0.7349	0.7416
90% Parent - 10% Leaf-nodes	Precision	0.7362	0.7418
	Recall	0.6973	0.7070
	F1	0.7061	0.7168

Table 4.6: Comparison of hierarchical classifier (HC) and Flat classifier on different data size of training data.

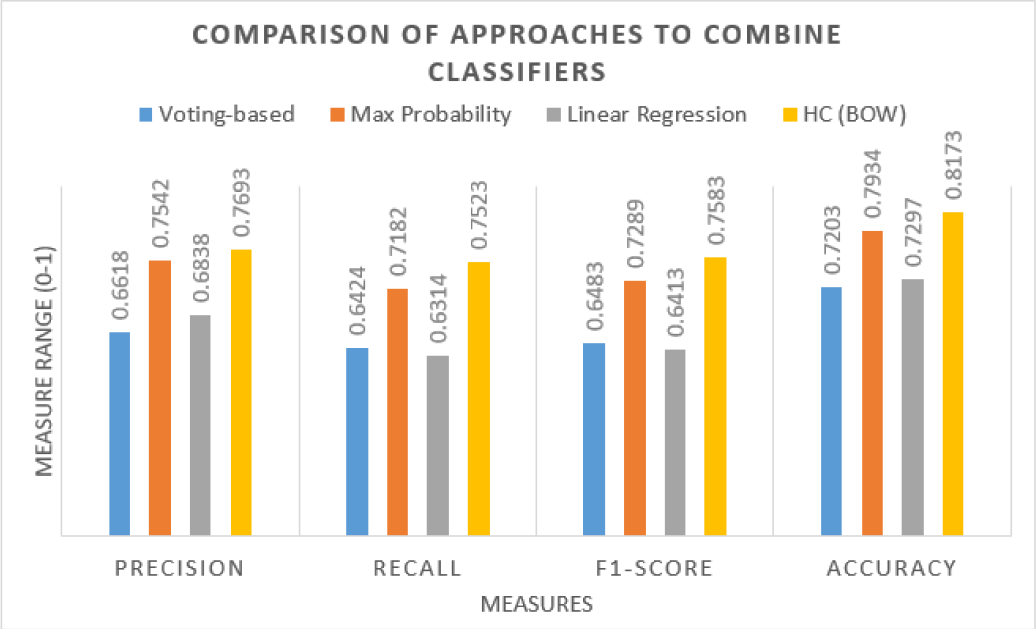


Figure 4.3: Comparison of flat classifier, hierarchical classifier based on path with maximum probability at each level and maximum of the probabilities from BOW, BOC and BOK hierarchical classifiers (macro average).

4.5 Consensus methods and the baseline model

For each representation, we have created a separate hierarchical classifier. Our aim to use a semantic hierarchical classifier is to improve the performance of hierarchical classifier based on BOW. The output from these three hierarchical classifiers needs to be combined to output one class. Our experiments used two simple arithmetic approaches and a Linear Regression. Two simple arithmetic approaches are the class with maximum probability from the output of hierarchical classifiers and the class with maximum number of the votes. The performance over 10 random runs of 70-30% train-test is taken into account and average of the results are reported in Figure 4.3 and 4.4. Though the performance of consensus method, the class with maximum probability, is competitive to hierarchical classifier based on BOW, the approaches to combine hierarchical classifier's output did not solve our problem to improve the performance of hierarchical classifier based on BOW.

To understand the problem of low performance on concepts and categories, the data for each evaluation groups individually and together is visualized using t-SNE

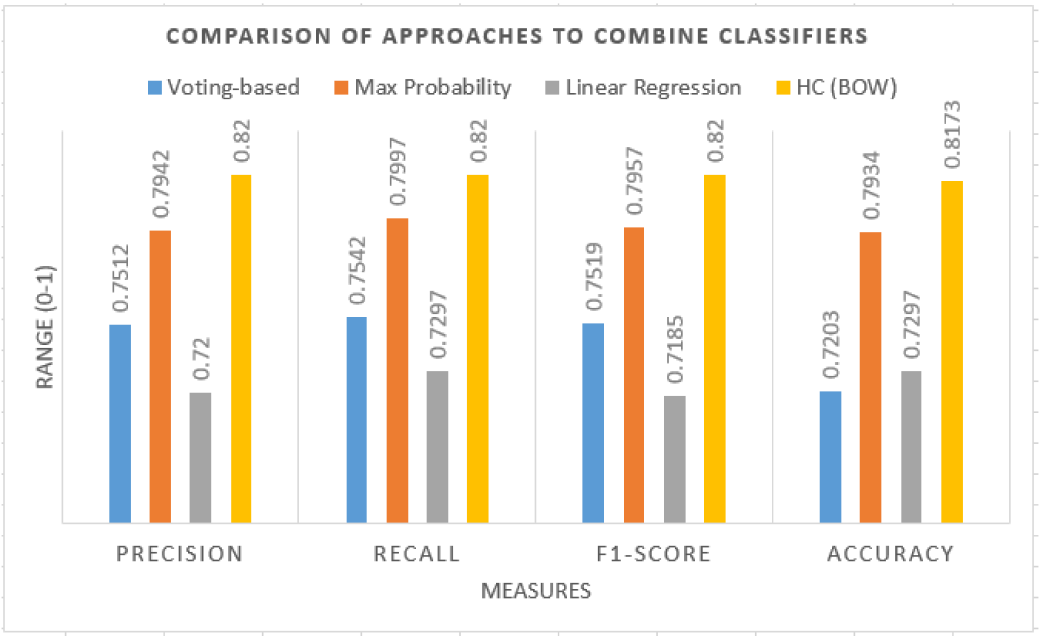


Figure 4.4: Comparison of flat classifier, hierarchical classifier based on path with maximum probability at each level and maximum of the probabilities from BOW, BOC and BOK hierarchical classifiers (weighted average).

(Maaten and Hinton, 2008). t-SNE uses Stochastic Neighbor Embedding to convert Euclidean distances between data points into conditional probability based on similarities (Maaten and Hinton, 2008).

The visualization using t-SNE for 3 evaluation groups for each type of document representations, BOW, BOC and BOK , is shown in Figure 4.5, 4.6 and 4.7 and for individual evaluation groups for each document representation in Appendix Section D. Each of these t-SNE Figures shows different color clusters with numbers which represents the research topic's label. The label for LSA start from 0-10, LSB from 11-20 and CS from 21-41. The clusters in Figure 4.5 have a dense overlapping of data points at few places, but most of them are separable from each other. Even in each cluster data points are not overlapping to form concentrated spot. The similar plot of data points for concepts vectors plotted using t-SNE (refer Figure 4.6) and many dense clusters with overlapping data points are noticed. These overlapping points suggest that they are very similar to each other and possibly hide many points from other classes. Similar observation for vectors of categories in Figure 4.7 could be

seen. The data points of concepts and categories overlap more than the words. The concepts and categories in the corpus of documents have more similarities compared to BOW and clutter on each other to create dense clusters. Therefore BOC and BOK for a document combine with other classes which result in less discrimination by a O-v-R classifier to accurately identify the correct class. The performance of different hierarchical classifiers and consensus methods do not help for the very same reason.

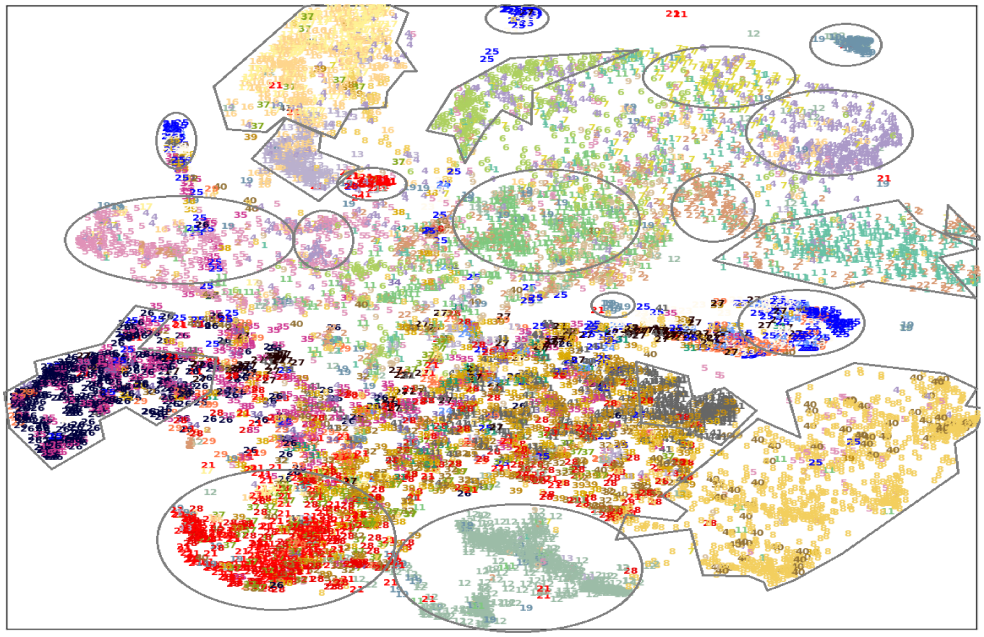


Figure 4.5: 2D plot of 6% samples from 3 evaluation groups on BOW using t-SNE.

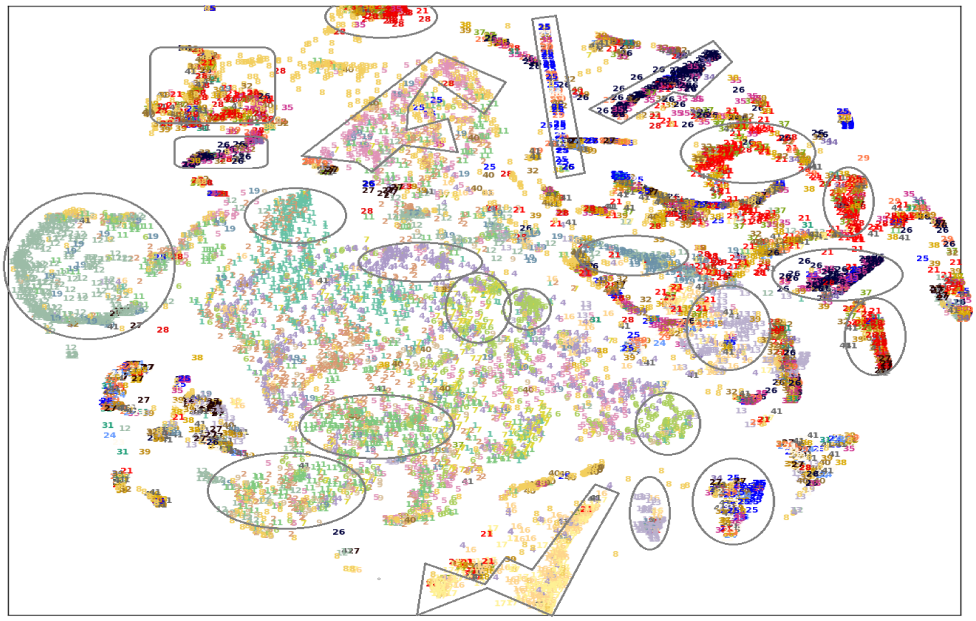


Figure 4.6: 2D plot of 6% samples from 3 evaluation groups on BOC using t-SNE.

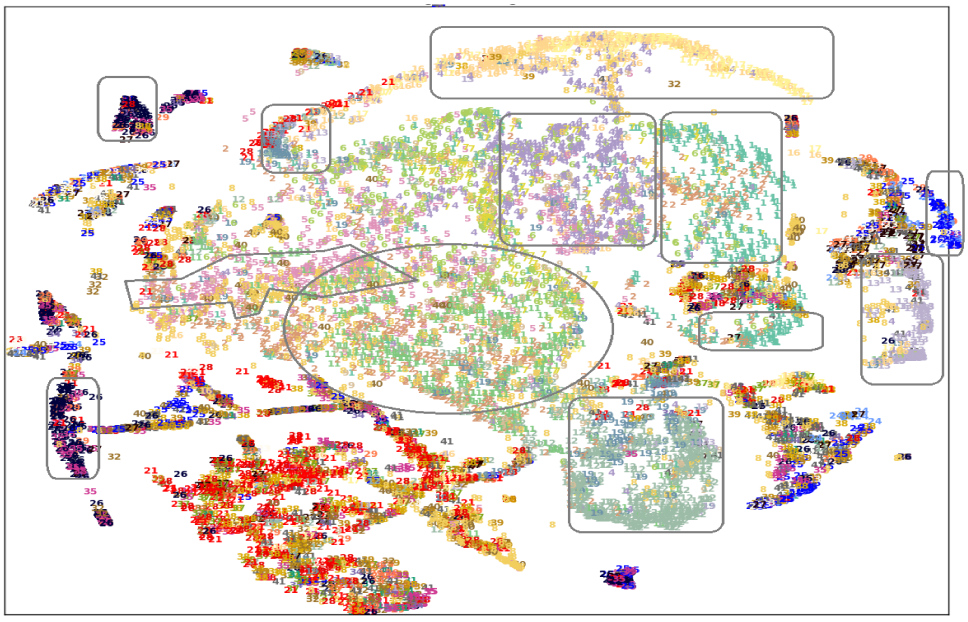


Figure 4.7: 2D plot of 6% samples from 3 evaluation groups on BOK using t-SNE.

4.6 Multi-labeling A Document

In reality, a scientific article can belong to multiple research topics and there is no definite label for such document. Therefore, this section of the thesis discusses the approaches to multi-label a document.

4.6.1 One-Class Classifier

A small dataset from the evaluation group LSA consisting of samples predicted incorrectly by multi-class O-v-R classifier with high probability is retrieved. From 64 of those documents, a small subset of randomly selected 14 documents were studied. First, each of these 14 documents were manually labeled and were assigned up-to two most matching labels of research topics by comparing the keywords defined by NSERC. The first preference is highlighted with bold and with underline for the second preference, if any, as shown in Table 4.7. We have used One-Class Classifier, (Schölkopf et al., 2000) and (Tax and Duin, 2004), using Linear SVM for each class for evaluation group LSA and each of the 14 documents were an input to One-Class SVM. One-Class SVM outputs positive or negative for a input document. Positive output means a document belongs to the research topic on which a One-Class SVM classifier is built. The results from three different approaches were studied to identify whether a document can belong to more than one research topic. Another reason to create such data set is to understand why O-v-R classifier predicted these documents incorrectly. The analysis of each document in the Table 4.7 is in Appendix.

The result from the Table 4.7 shows that O-v-R classifier predicts the class that is human-verified most of the times. The result in Table 4.7 shows that the One-Class classifiers does predict the original class and the human-verified labels (bold and underline), but it also predicts the classes that are incorrect. One-Class classifiers can be optimized to obtain optimal results but optimizing parameters for each research topics is hard. To trust the observation that there exists multiple labels for a document and similarity between document across research topics, more analysis on a larger data set is done using 10-fold cross validation and different parameters of

Index	Original	O-v-R Classifier	One-Class classifier (Positive Classes)
1526	LSA04	**LSA08 (0.40)**	LSA01, LSA04, **LSA08**, LSA05
2334	LSA07	**LSA01 (0.44)**	**LSA01**
2390	LSA05	**LSA08 (0.49)**	LSA01, LSA05, **LSA08**
3064	LSA05	**LSA08 (0.44)**	LSA01, LSA05, **LSA08**, LSA04
3607	LSA04	**LSA05 (0.47)**	LSA01, LSA04, **LSA05**
4105	LSA01	**LSA06 (0.41)**	LSA01, **LSA06**, LSA04, LSA05
5156	LSA01	**LSA06 (0.46)**	N/A
5331	LSA02	**LSA01 (0.40)**	**LSA01**, LSA06, LSA04, LSA05, LSA08, LSA09
6094	LSA07	**LSA02 (0.50)**	LSA01, **LSA02**, LSA04, LSA03, LSA06, LSA07, LSA08, LSA05, LSA09
6450	LSA01	**LSA05 (0.43)**	LSA01, **LSA05**, LSA08, LSA04, LSA06
8541	**LSA09**	LSA05 (0.40)	LSA01, LSA04, LSA05
9414	LSA05	**LSA08 (0.42)**	**LSA08**
14088	LSA06	**LSA01 (0.50)**	**LSA01**, LSA06, LSA03, LSA04, LSA09
19831	**LSA08**	LSA01 (0.42)	LSA01

Table 4.7: 14 documents predicted incorrectly with high probability by O-v-R classifier and comparison with manually labeled (bold and underline) and One-Class SVM output for parameter nu=0.1.

One-Class SVM. It is expected that One-Class classifier will have the least error on the data from the same class and largest on the other classes. The results from each One-Class classifier can be found in Figure 4.8 and in Appendix E.1. The result in the Figure shows that research topic LSA01 is more closely related to LSA02 and LSA07 than other research topics. A similar observations are visible with other One-Class classifiers.

One can use One-Class SVM to find the correct classes to multi-label a document, but to select an optimal parameters for each research topic on which it is trained is hard. Therefore, the probabilities from O-v-R classifier is further explored in the next section to multi-label a document.

[3]One-Class SVM: http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

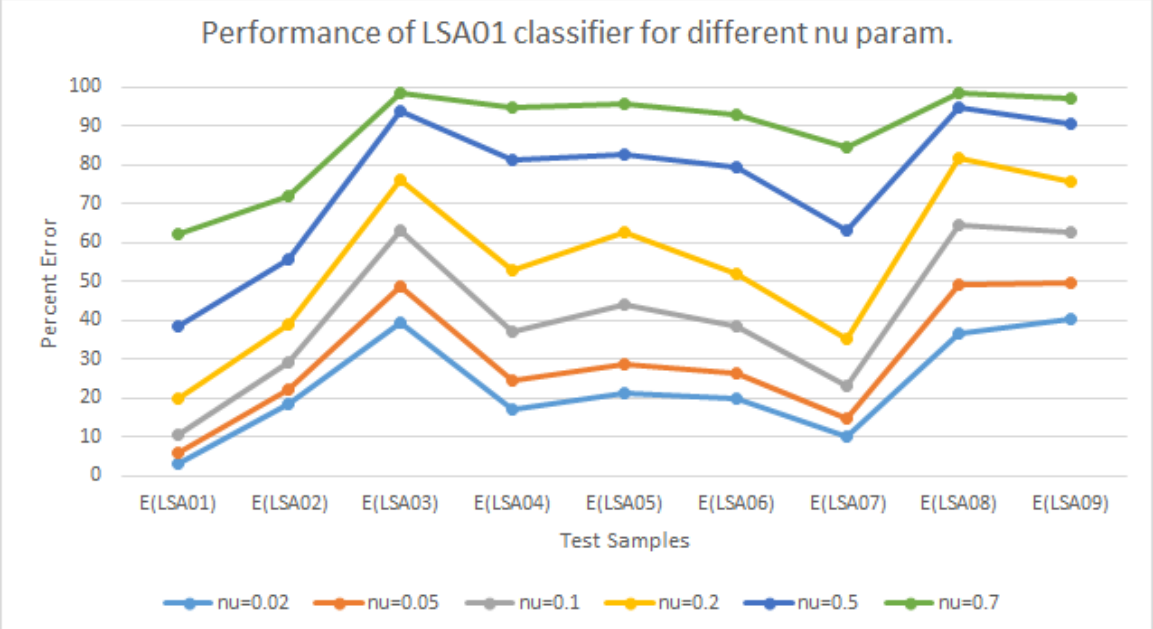


Figure 4.8: 10-Fold CV of One-Class classifier for evaluation group LSA01 on various 'nu' parameter shows that research topics LSA07 and LSA02 are similar to LSA01 on which the classifier is trained.

4.7 Choosing labels and Multi-labeling

In this section, we discuss on how top n classes from O-v-R classifiers affect the chance of having original label and provides evidence for the scope to multi-label a document. We have discussed two approaches, threshold-based approach mitigates the drawback of local classifiers whereas other method combines the output from consensus methods.

One class classifier have better performance on most of the classifier except neural net (Manevitz and Yousef, 2002), but parameter selection is challenging when it is sensitive to the parameters (Xiao et al., 2015). Due to challenges posed by One-Class classifier to get an optimal results, we have explored O-v-R classifier for the potential to multi-label a document. The probabilities from O-v-R classifier for evaluation group LSA and CS were visualized using histogram and normal curve to understand the spread of probabilities of top n class(es) by probabilities. O-v-R classifiers discriminate between classes based on statistics of the terms and therefore they are sensitive to the use of terms in a document. If there are terms that span multiple

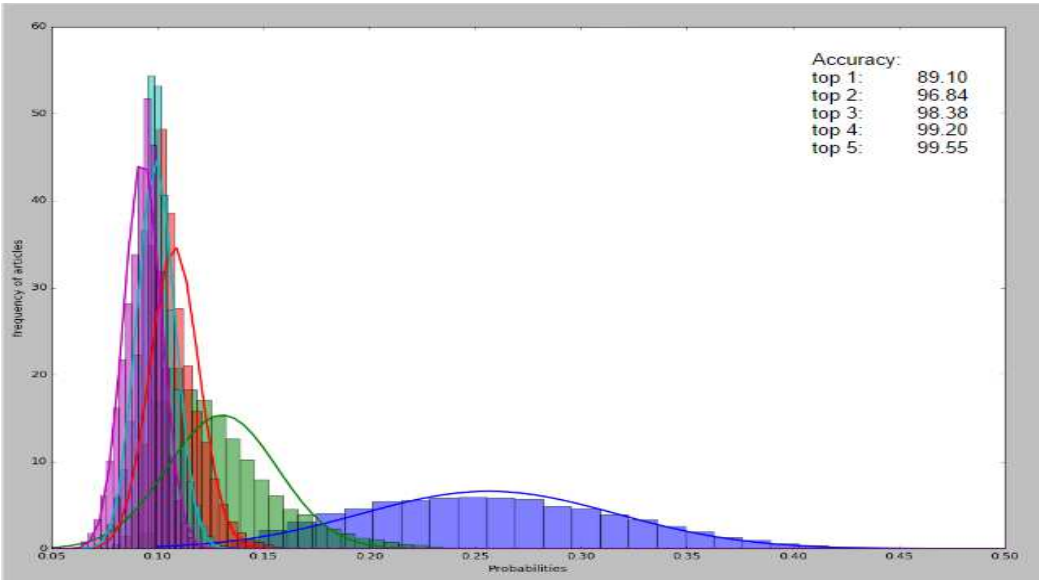


Figure 4.9: Normal distribution of top n predicted probabilities for each class by O-v-R classifier on all the test samples from the evaluation group LSA. The distribution of probabilities for the predicted class is shown in blue and followed by second highest and third highest probability distribution and so on. On the top right corner is the probability/accuracy of having an original label in top n predicted classes.

classes then based on some threshold value top n class(es) can be selected. To investigate how much top n class(es) from O-v-R classifier contributes to the probability or accuracy of having an original label, a 10-fold cross validation is performed on evaluation group LSA and CS and the results are in top right corner of Figure 4.9 and 4.10. The result shows that the accuracy of having label in top n probabilities sharply increases and then gradually decreases. Similar observation is visualized using histogram and normal curve plot where the distribution of n^{th} order probabilities from O-v-R classifier shows the fall in the spread of probabilities and overlaps each other. In contrast, the distribution of top 3 probabilities shows visible separation of normal curve with some overlap with each other. The normal curves overlaps as there is a gradual decrease in accuracy of having a class in top n predicted by O-v-R classifier.

The problem with choosing a threshold is that the hierarchical classification may stop at the top level or intermediate level if threshold value is not satisfied. This results in a blocking problem where the hierarchical classification does not research the leaf-node. So to avoid the blocking problem, we have used $1/k$ as the threshold

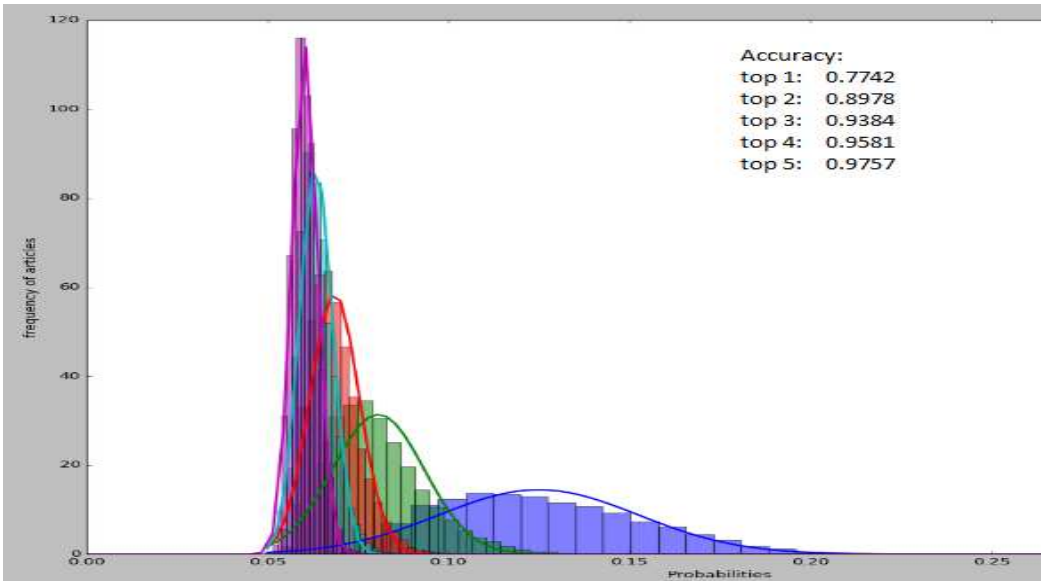


Figure 4.10: Normal distribution of top n predicted probabilities for each class by O-v-R classifier on all the test samples from an evaluation group CS. The distribution of probabilities for the predicted class is shown in blue and followed by second highest and third highest probability distribution and so on. On the top right corner is the probability/accuracy of having an original label in top n predicted classes.

where k is the number of classes of each parent node. For an example in Figure 4.11, at the first level we have used 1/3 threshold-value to predict one or more nodes. The same rule can be applied to the second level of a parent node LSA where $k = 8$ and other parent nodes that pass threshold values in the previous level. One can choose to predict top n probabilities for research topic or the one with maximum probability. In our proposed methodology we have used threshold at the top-level to select the nodes and the class with maximum probability at leaf-level for each of these nodes. The predicted class or classes from different hierarchical classifiers and for each path in hierarchy based on threshold-value are combined to finally output up-to top 3 classes based on votes (Figure 4.12).

In multi-label process, it is important to control the number of documents in the data set that are multi-labeled because not all research articles belongs to multiple research topics. Therefore, another approach is used that combines output from consensus methods of proposed methodology and output up-to top 3 classes as shown

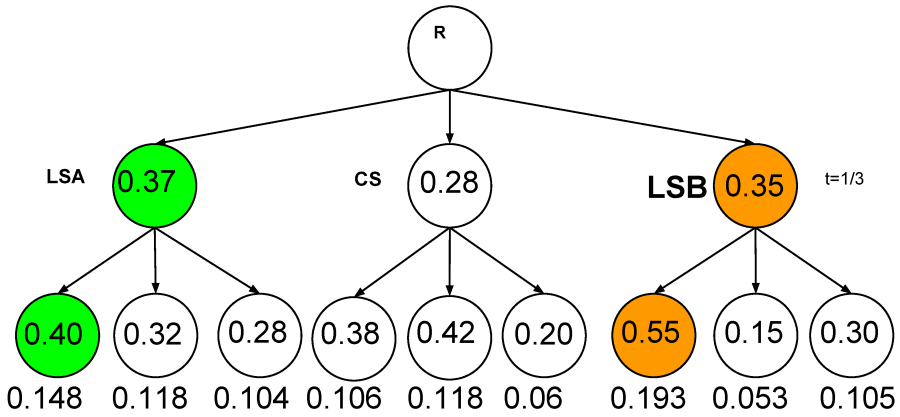


Figure 4.11: Multiple paths selected using threshold value at the top-level (evaluation groups).

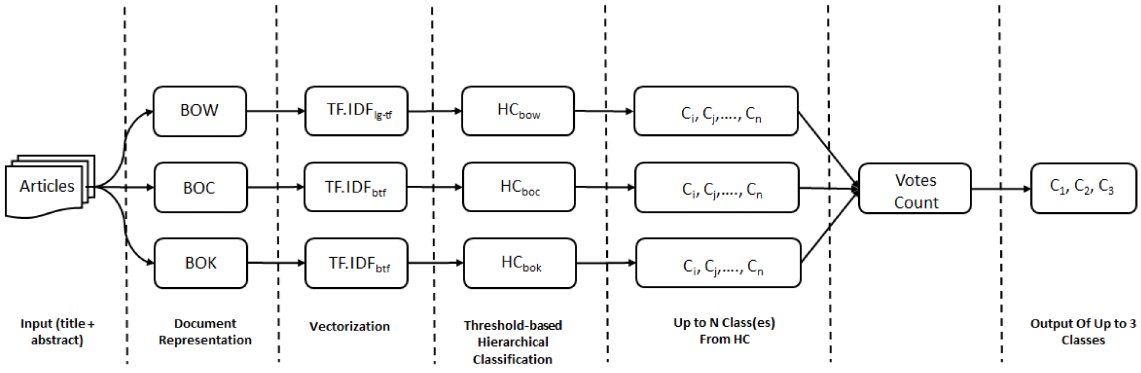


Figure 4.12: Threshold based hierarchical classification that outputs multiple classes and then selecting top 3 classes based on votes count.

in Figure 3.1. The normal distribution of top n probabilities shows that after 3^{rd} order probabilities from O-v-R classifier on evaluation group LSA and CS, the normal graph are similar to each other and they overlap. Though distribution of 3^{rd} order probabilities shows little difference from distribution of probabilities after 3^{rd} order, there are significant number of articles with three labels. This does not mean that there cannot be an article with more than three labels, but the number is insignificant compared to articles with three labels.

Threshold based approach to multi-label a document					
	Hierarchical Classifier	Threshold based HC	1-label	2-labels	3-labels
	Accuracy	Accuracy	# of docs	# of docs	# of docs
BOW	81.74	84.28	45,871	5,521	0
BOC	67.76	71.26	44,258	7,134	0
BOK	66.98	70.72	43,742	7,650	0
BOW + BOC + BOK	79.34	88.84	28,095	17,126	6,171

Consensus method to multi-label a document					
	Hierarchical Classifier	Consensus method	1-label	2-labels	3-labels
	Accuracy	Accuracy	# of docs	# of docs	# of docs
BOW	81.74	82.69	49,877	1,515	0
BOC	67.76	69.43	48,582	2,810	0
BOK	66.98	68.57	48,575	2,817	0
BOW + BOC + BOK	79.34	82.11	43,249	8,143	0

Table 4.8: Performance of two proposed approaches applied on different hierarchical classifiers based on document representations.

The results in Table 4.8 shows number of research articles multi-labeled using threshold-based approach on hierarchical classifiers based on BOW, BOC and BOK. Using this approach about 45% of the research articles were multi-labeled for an increase of about 8% of of having original label in predicted label(s). In contrast, consensus method multi-labels fair number of research article, but it also does poor

compared to threshold based approach. About 8% articles were multi-labeled for an increase of about 2.5% of having original label in predict label(s). Though both approaches have drawback, but the latter approach controls the chances of false positive rates. It is worth not multi-labeling a research article rather than adding incorrect labels.

The potential weaknesses and strengths of approaches to multi-label a document are investigated using few documents. Each document is concatenated using the title and an abstract where first line is the title of the document. Consider text example in below text box, which belongs to the research topic LSA07 (Cell Signals and Electrical Properties) in our data set. The text talks about research topic LSA01 (Immunology) and the research problem is addressed in research topic LSA07. Though manual verification assigns such document as LSA01 and LSA07, but it is challenging for O-v-R model to label it as LSA07 because of sparse use of terminology from LSA07. Threshold-based method and consensus method also predict the research topic LSA01 as expected. Both the methods performed the same in this case.

The T-Cell Antigen Receptor: A Logical Response to an Unknown Ligand. *The immune system can be roughly divided into innate and adaptive compartments. The adaptive compartment includes the B and T lymphocytes, whose antigen receptors are generated by recombination of gene segments. The consequence is that the creation of self-reactive lymphocytes is unavoidable. For the host to remain viable, the immune system has evolved a strategy for removing autoimmune lymphocytes during development. This review discusses how T lymphocytes are generated, how they recognize antigens, and how their antigen receptor directs the removal of self-reactive T cells.*

It has been observed that if a research article belongs to research topics across evaluation groups then hierarchical classification performs better in multi-labeling a document by selecting multiple paths based on threshold value. However, multi-labeling using consensus methods often ignore such cases. The text is an example of research topic LSA05 (Molecular Genetics), but the research problem is addressed by

LSA08 (Quantitative Approaches). This text more specifically talks about Bioinformatics which also belongs to research topic CS20 (Bioinformatics and Bio-inspired Computing) from Computer Science evaluation group. The decision to choose multi-paths is only possible in threshold-based approach and therefore threshold based approach multi-labeled this document as LSA05, LSA08 and CS20 whereas consensus based approach labeled it as LSA05 and CS20.

Software for constructing and verifying pedigrees within large genealogies and an application to the Old Order Amish of Lancaster County. *This paper describes PedHunter, a software package that facilitates creation and verification of pedigrees within large genealogies. A frequent problem in medical genetics is to connect distant relatives with a pedigree. PedHunter uses methods from graph theory to solve two versions of the pedigree connection problem for genealogies as well as other pedigree analysis problems. The pedigrees are produced by PedHunter as files in LINKAGE format ready for linkage analysis. PedHunter uses a relational database of genealogy data, with tables in specified format, for all calculations. The functionality and utility of PedHunter are illustrated by examples using the Amish Genealogy Database (AGDB), which was created for the Old Order Amish community of Lancaster County, Pennsylvania.*

An example from evaluation group CS is shown in below text box, where the document is originally labeled and human verified as CS18 (Artificial Intelligence). Since this text belongs to CS18, it is correctly classified as single label by consensus method approach. But, threshold-based method takes multi-paths because of the use of terms that describes LSA08 (Quantitative Approaches) as well as CS18. It also classifies the text into CS01 (Web Enabled Services and Applications). The multi-labels using threshold-based method has incorrect labels in this case.

Bootstrapping a Game with a Purpose for Commonsense Collection. *Text mining has been very successful in extracting huge amounts of commonsense knowledge from data, but the extracted knowledge tends to be extremely noisy. Manual construction of knowledge repositories, on the other hand, tends to produce high-quality data in very small amounts. We propose an architecture to combine the best of both worlds: A game with a purpose that induces humans to clean up data automatically extracted by text mining. First, a text miner trained on a set of known commonsense facts harvests many more candidate facts from corpora. Then, a simple slot-machine-with-a-purpose game presents these candidate facts to the players for verification by playing. As a result, a new dataset of high precision commonsense knowledge is created. This combined architecture is able to produce significantly better commonsense facts than the state-of-the-art text miner alone. Furthermore, we report that bootstrapping (i.e., training the text miner on the output of the game) improves the subsequent performance of the text miner.*

Another example is from research topic CS11 (Programming Languages) (refer below text box) where it is human-verified as CS15 (Parallel and Distributed Computing) and also by threshold-based and consensus methods. However, due to multi-paths by threshold-based approach, the classifier predicts LSB07 (Cognitive Science) which is an incorrect label. It is important to note that even though the original label is not related to CS11 (Programming Languages), classifier is able to detect the correct research topic. The original label CS15 is because this article is accepted by the journal that is not related to CS15. Such articles can be re-assigned to correct journals by the publications.

From all these examples it is learned that both the approaches have advantages and drawbacks and therefore there is scope for novel approaches that accurately multi-label a document.

Exploiting reference idempotency to reduce speculative storage overflow. *Recent proposals for multithreaded architectures employ speculative execution to allow threads with unknown dependences to execute speculatively in parallel. The architectures use hardware speculative storage to buffer speculative data, track data dependences and correct incorrect executions through roll-backs. Because all memory references access the speculative storage, current proposals implement speculative storage using small memory structures to achieve fast access. The limited capacity of the speculative storage causes considerable performance loss due to speculative storage overflow whenever a thread's speculative state exceeds the speculative storage capacity. Larger threads exacerbate the overflow problem but are preferable to smaller threads, as larger threads uncover more parallelism.In this article, we discover a new program property called memory reference idempotency. Idempotent references are guaranteed to be eventually corrected, though the references may be temporarily incorrect in the process of speculation. Therefore, idempotent references, even from nonparallelizable program sections, need not be tracked in the speculative storage, and instead can directly access nonspeculative storage (i.e., conventional memory hierarchy). Thus, we reduce the demand for speculative storage space in large threads. We define a formal framework for reference idempotency and present a novel compiler-assisted speculative execution model. We prove the necessary and sufficient conditions for reference idempotency using our model. We present a compiler algorithm to label idempotent memory references for the hardware. Experimental results show that for our benchmarks, over 60% of the references in nonparallelizable program sections are idempotent.*

Chapter 5

Conclusion & Discussion

In our research, we used LCPN, a type of hierarchical classification approach, to classify research articles into one or more sub-categories/classes. We have implemented separate multiclass (O-v-R) classifier for all research topics of each evaluation group defined by NSERC and then created LCPN hierarchical classifier model. We elicited titles and abstract from journals that are a close match to the keywords provided by NSERC research topics. Each document, comprised of title and abstract, is represented as BOW, BOC and BOK.

A hierarchical classifier is trained using Linear SVM and the performance of O-v-R classifier is tested on different document representation. It is observed that there is a drastic difference in performance on different document representation such as BOC and BOK. Similar results of performance of hierarchical classifiers on BOC and BOK are observed. A 2-Dimensional (2D) t-SNE plots are used to visualize the relationships or similarities between different data points. The clusters in t-SNE plot for BOW shows that most of the clusters are separated from each other with some noise in each cluster. The noise is any other label that does not belong to the label the cluster represents. The data points in each clusters are also separated from each other compared to 2D t-SNE plots of BOC and BOK. If the data points clutter on each other to create the concentrated spot then those points have no distinguishing characteristics. These dense clusters may hide other labels which are not visible. Therefore the performance of hierarchical classifiers and O-v-R classifiers on BOC and BOK have poor performance compared to BOW.

A local hierarchical classifier of type LCPN is trained on 70% of titles and abstracts of the articles using "exclusive sibling" policy. We proposed two methods to

predict the class. The first method is based on the path with maximum probability and other is the path with maximum product of the probabilities. The blocking problem posed by local classifier could be resolved using product of the probabilities, but this itself is susceptible to imbalance in the number of nodes which significantly affect the performance of hierarchical classifier.

The performance of LCPN classifiers for each document representation are compared with flat classifiers for the same document representation. The performance of the hierarchical classifier is competitive to the flat classifiers, but the performance of the latter degrades when there are many classes and less number of samples to train.

The primary purpose of different document representations such as BOC and BOK was to enrich BOW and performance over BOW. The predicted classes and probabilities are combined using consensus methods to output a single class. It was expected that the consensus methods will improve the performance over BOW. Though the performance of consensus method, class with maximum probability, is competitive, but all the methods failed to improve the performance over hierarchical classifier based on BOW.

Second contribution to the research is to multi-label a document. Two methods were proposed to multi-label a document. A threshold-based approach is used at the top-level to select multiple paths if more than one classes have equal probabilities. The threshold value of 1/k is used and it has been observed that it increased the accuracy of having an original label in the multi-label but at the cost of proportion of documents being multi-labeled. For an increase of 9% of accuracy about 45% of the documents were multi-label. Therefore another method is used to control the number of documents being multi-labeled. The output from consensus method is combined to multi-label a document. It performs better than threshold-based approach, but it fails to identify the research topics across the evaluation groups because it does not considered multiple paths.

NSERC research topics are defined by keywords which have sub-topics and such

keywords can be further divided into sub-topics of a research topic. There are fine grained research topics such as LSA03 (Organelle Function and Intracellular Trafficking) and LSA10 (Cell Cycle) and could be part of other research topics. LSA10 (Cell Cycle), for an example, is very common topic and terms from articles in this research topic are shared across multiple research topics. The challenge here is in finding enough articles on such research topics that are inclusive and have enough terms to discriminate across research topics. This problem is not limited to an evaluation group, but across evaluation groups. The research topic LSA08 (Quantitative approach) has a keyword Bioinformatics which is a research topic CS20 (Bioinformatics and Bio-inspired Computing) in Computer Science. The articles on Bioinformatics have poor performance because the hierarchical classifier fails to discriminate between LSA or CS evaluation group at the top level and error is propagated down the hierarchy.

5.1 Future Work

There are different hierarchical classification approaches, such as local and global, where local classifiers that completely ignores the class hierarchy suffers from blocking problem, and are prone to inconsistency which can be defeated by using global classifier (Silla and Freitas, 2010). The global classifier is complex to train and therefore very less amount of research is done using it.

The current implementation of wikification using Wikipedia can be replaced with Wikifier by University of Illinois (Ratinov et al., 2011) (Cheng and Roth, 2013). They have optimized disambiguation task using local and global approaches and proposed a new global system, GLOW, that outperforms the state-of-the-art system of (Milne and Witten, 2008). Experiments on different dataset have shown significant improvement in detecting more concepts. Disambiguation task using Wikipedia relies on local and global statistics for a given text to find candidate Wikipedia articles. These approaches use Wikipedia articles and link structure which often fails to address the underlying contextual knowledge. This problem has been potentially resolved

by (Cheng and Roth, 2013) in Illinois Wikifier[1] software implementation. Though there could be a problem to find corresponding categories in Wikipedia for concepts detected by Illinois Wikifier because it uses external data sources to detect concepts missed by our use of Wikipedia Miner toolkit for Wikification. Nevertheless, there is a chance to improve our work in spite of mentioned shortcomings.

In our implementation, concepts are treated as separate from BOW and binary TF.IDF VSM is created for concepts. One can think of combining these concepts with BOW by replacing it with mentions/terms representing a concept.

There can be concepts/terms in the text which have no direct relationship with other terms such as protein-to-protein, and protein-to-disease-to-diagnosis, which may have been missed during wikification. One way to resolve this issue is by using Concept Chain Queries (CCQ)s proposed by (Jin and Srihari, 2006). Their research focuses on identifying relationships between two terms by chaining related documents from the corpus and reaching to potential relationship conclusion. However, it relies on a software tool to detect entities and terms in documents between which the relationships can be queried. This problem is then addressed by (Yan and Jin, 2012) using Explicit Semantic Analysis technique (ESA) originally proposed by (Gabrilovich and Markovitch, 2007) and (Jin and Srihari, 2006) on BOW. This BOW were converted to weighted score measured by its TF.IDF score. In most recent work by (Yan and Jin, 2016), they have incorporated Wikipedia concepts and categories to find relationship between two text, if no relation is discovered using original implementation of CCQ.

There are various semantic kernels built in the past to overcome the limitations of kernels based on BOW representations and future work in this direction can improve the performance of classification model to predict correct class for a given research document/article. One of the research by (Altınel et al., 2015) has customized the linear kernel to take advantage of background knowledge which uses Helmholtz principle from Gestalt theory to measure meaningfulness of terms in the context of classes. These document measure vectors can be created using external sources such

[1]https://cogcomp.cs.illinois.edu/page/software_view/Wikifier

as Wikipedia and incorporated into semantic kernel defined in their work. The results from their experiments have shown to improve results than traditional classification algorithms.

Another approach is by combining all of these above-mentioned work by first using Illinois Wikifier to extract concepts and then identify most likely articles/topics in Wikipedia and other latent relationship between concepts using CCQs. After detecting these concepts and creating vector space model of these concepts, identify categories for each of these concepts and create a bag of categories. Finally apply Helmholtz principle to extract meaningful terms from BOW, BOC, and BOK and then implement semantic kernel. One can also consider implementing above-mentioned semantic kernels and evaluate the performance of each of these kernels.

In this entire research, it is important to critic on the dataset. Chosen evaluation group and journals for corresponding research topics were solely on researchers' knowledge, and therefore cross verification is required to ensure all keywords of research topics are covered. One can choose to do clustering on each evaluation group and create clusters equal to research topics. The keywords from these separable clusters can be used to create weighted TF.IDF vectors or weighted Perceptron for BOW, BOC, and BOK and these models can then be evaluated with our work.

The performance of two approaches to mutli-label a document can be improved by using the distribution of probabilities of different classes and making a decision to choose one or more classes. A research belongs to research topic within an evaluation group than across the evaluation groups and therefore distribution of probabilities can be applied to the hierarchical classifier to select one or more nodes. Depending on the distribution of the probabilities from O-v-R classifier, the articles with more than three labels can be selected accurately.

Dataset created by targeting journal as a data source proved to be very useful and have the potential for data mining and machine learning applications. In our work, we have focused on three evaluation groups and research topics for which enough

documents could be extracted. In the future work, it can be applied to all twelve evaluation groups defined by NSERC. Use of clustering to identify additional keywords for each research topics is likely to improve the performance of hierarchical classifier. One of the applications of this work is to identify and rank expertise of a researcher. Under an assumption from our findings that a researcher's interest generally do not span across multiple evaluation groups and not more than three research topics at a given time, the output from hierarchical classifier on documents of a researcher can be combined with documents predicted into the same class and could be a feedback to the hierarchical classifier. Then, the correct class can be identified from the probabilities by the hierarchical classifier. Again, top two or three probabilities of each document of a researcher from the hierarchical classifier can be combined arithmetically and research topics can be ranked.

Finally, this research can be extended to create a profile for each researcher. The research articles of a researcher can be feed to the trained classifier using our proposed methodology and based on the predicted research topics for each document, a profile of a researcher can be created.

Bibliography

Altinel, B., Ganiz, M. C., and Diri, B. (2014). A semantic kernel for text classification based on iterative higher–order relations between words and documents. In *Artificial Intelligence and Soft Computing*, pages 505–517. Springer Science + Business Media.

Altınel, B., Ganiz, M. C., and Diri, B. (2015). A corpus-based semantic kernel for text classification by using meaning values of terms. *Engineering Applications of Artificial Intelligence*, 43:54–66.

Altlnel, B., Ganiz, M. C., and Diri, B. (2013). A novel higher-order semantic kernel for text classification. In *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, pages 216–219. Institute of Electrical and Electronics Engineers (IEEE).

Alves, R. T., Delgado, M. R., and Freitas, A. A. (2008). Multi-label hierarchical classification of protein functions with artificial immune systems. In *Proceedings of the 3rd Brazilian Symposium on Bioinformatics: Advances in Bioinformatics and Computational Biology*, BSB '08, pages 1–12, Berlin, Heidelberg. Springer-Verlag.

Banerjee, S., Ramanathan, K., and Gupta, A. (2007). Clustering short texts using wikipedia. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 787–788, New York, NY, USA. ACM.

Beel, J., Gipp, B., Langer, S., and Breitinger, C. (2016). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338.

Borges, H. B., Silla, C. N., and Nievola, J. C. (2013). An evaluation of global-model hierarchical classification algorithms for hierarchical classification problems with single path of labels. *Computers & Mathematics with Applications*, 66(10):1991–2002.

Budanitsky, A. and Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Cerri, R., Barros, R. C., and de Carvalho, A. C. (2014). Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80(1):39–56.

Cerri, R., Barros, R. C., and de Carvalho, A. C. P. L. F. (2011). Hierarchical multi-label classification for protein function prediction: A local approach based on neural

networks. In *2011 11th International Conference on Intelligent Systems Design and Applications*, pages 337–343.

Cerri, R., Barros, R. C., P. L. F. de Carvalho, A. C., and Jin, Y. (2016). Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, 17(1):373.

Cerri, R. and de Carvalho, A. C. P. L. F. (2011). Hierarchical multilabel protein function prediction using local neural networks. In *Advances in Bioinformatics and Computational Biology*, pages 10–17. Springer Science + Business Media.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. (2006). Hierarchical classification: Combining bayes with svm. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 177–184, New York, NY, USA. ACM.

Charlin, L., Zemel, R. S., and Boutilier, C. (2012). A framework for optimizing paper matching. *CoRR*, abs/1202.3706.

Chen, H.-H., Treeratpituk, P., Mitra, P., and Giles, C. L. (2013). Csseer: An expert recommendation system based on citeseerx. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, pages 381–382, New York, NY, USA. ACM.

Cheng, X. and Roth, D. (2013). Relational inference for wikification. In *EMNLP*.

D'Alessio, S., Murray, K., Schiaffino, R., and Kershenbaum, A. (2000). The effect of using hierarchical classifiers in text categorization. In *Content-Based Multimedia Information Access - Volume 1*, RIAO '00, pages 302–313, Paris, France, France. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Deng, H., King, I., and Lyu, M. R. (2008). Formal models for expert finding on dblp bibliography data. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 163–172, Washington, DC, USA. IEEE Computer Society.

Dumais, S. and Chen, H. (2000). Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 256–263, New York, NY, USA. ACM.

Esuli, A., Fagni, T., and Sebastiani, F. (2008). Boosting multi-label hierarchical text categorization. *Inf. Retr.*, 11(4):287–313.

Gabrilovich, E. and Markovitch, S. (2006). Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1301–1306. AAAI Press.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The Twentieth International Joint Conference for Artificial Intelligence*, pages 1606–1611, Hyderabad, India.

Guan, Y., Myers, C. L., Hess, D. C., Barutcuoglu, Z., Caudy, A. A., and Troyanskaya, O. G. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biol*, 9(Suppl 1):S3.

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. In *2008 Fourth International Conference on Natural Computation*, volume 4, pages 192–201.

Hernández, J., Sucar, L. E., and Morales, E. F. (2014). Multidimensional hierarchical classification. *Expert Systems with Applications*, 41(17):7671–7677.

Holden, N. and Freitas, A. A. (2008). *Improving the Performance of Hierarchical Classification with Swarm Intelligence*, pages 48–60. Springer Berlin Heidelberg, Berlin, Heidelberg.

Hu, J., Fang, L., Cao, Y., Zeng, H.-J., Li, H., Yang, Q., and Chen, Z. (2008). Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 179–186, New York, NY, USA. ACM.

Hu, X., Zhang, X., Lu, C., Park, E. K., and Zhou, X. (2009). Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 389–396, New York, NY, USA. ACM.

Huang, A., Milne, D., Frank, E., and Witten, I. H. (2009). *Clustering Documents Using a Wikipedia-Based Concept Representation*, pages 628–636. Springer Berlin Heidelberg, Berlin, Heidelberg.

Isenberg, P., Isenberg, T., Sedlmair, M., Chen, J., and Möller, T. (2014). Toward a deeper understanding of visualization through keyword analysis. *CoRR*, abs/1408.3297.

Jin, B., Muller, B., Zhai, C., and Lu, X. (2008). Multi-label literature classification based on the gene ontology graph. *BMC Bioinformatics*, 9(1):525.

Jin, W. and Srihari, R. K. (2006). Knowledge discovery across documents through concept chain queries. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, ICDMW '06, pages 448–452, Washington, DC, USA. IEEE Computer Society.

Jing, L., Ng, M. K., and Huang, J. Z. (2010). Knowledge-based vector space model for text clustering. *Knowledge and Information Systems*, 25(1):35–55.

Levatić, J., Kocev, D., and Džeroski, S. (2014). The importance of the label hierarchy in hierarchical multi-label classification. *J Intell Inf Syst*, 45(2):247–271.

Lipczak, M., Koushkestani, A., and Milios, E. (2014). 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, New York, NY, USA. ACM.

Liu, T.-Y., Yang, Y., Wan, H., Zeng, H.-J., Chen, Z., and Ma, W.-Y. (2005). Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, 7(1):36–43.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Madjarov, G., Gjorgjevikj, D., Dimitrovski, I., and Džeroski, S. (2016). The use of data-derived label hierarchies in multi-label classification. *J Intell Inf Syst*, 47(1):57–90.

Manevitz, L. M. and Yousef, M. (2002). One-class svms for document classification. *J. Mach. Learn. Res.*, 2:139–154.

Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Scoring, term weighting, and the vector space model*, page 100123. Cambridge University Press.

Mavroeidis, D. (2005). *Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification*, pages 181–192. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mihalcea, R. and Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA. ACM.

Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA. ACM.

Milne, D. and Witten, I. H. (2013). An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194:222–239.

Moreira, C., Calado, P., and Martins, B. (2013). Learning to rank academic experts in the DBLP dataset. *Expert Systems*, 32(4):477–493.

Nasir, J. A., Karim, A., Tsatsaronis, G., and Varlamis, I. (2011). *A Knowledge-Based Semantic Kernel for Text Classification*, pages 261–266. Springer Berlin Heidelberg, Berlin, Heidelberg.

Poyraz, M., Kilimci, Z. H., and Ganiz, M. C. (2014). Higher-order smoothing: A novel semantic smoothing method for text classification. *J. Comput. Sci. Technol.*, 29(3):376–391.

Puurula, A., Read, J., and Bifet, A. (2014). Kaggle LSHTC4 winning solution. *CoRR*, abs/1405.0546.

Ramírez-Corona, M., Sucar, L. E., and Morales, E. F. (2016). Hierarchical multilabel classification based on path evaluation. *International Journal of Approximate Reasoning*, 68:179–193.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA. Association for Computational Linguistics.

Riahi, F., Zolaktaf, Z., Shafiei, M., and Milios, E. (2012). Finding expert users in community question answering. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 791–798, New York, NY, USA. ACM.

Schölkopf, P. B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (2000). Support vector method for novelty detection. In Solla, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12*, pages 582–588. MIT Press.

Secker, A., Davies, M., Freitas, A., Clark, E., Timmis, J., and Flower, D. (2010). Hierarchical classification of g-protein-coupled receptors with data-driven selection of attributes and classifiers. *International Journal of Data Mining and Bioinformatics*, 4(2):191.

Secker, A. D., Davies, M. N., Freitas, A. A., Timmis, J., Mendao, M., and Flower, D. R. (2007). An experimental comparison of classification algorithms for hierarchical prediction of protein function. *Expert Update (Magazine of the British Computer Society's Specialist Group on AI)*, 9(3):17–22.

Silla, C. N. and Freitas, A. A. (2009). Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 3499–3504.

Silla, C. N. and Freitas, A. A. (2010). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Sun, A. and Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 521–528, Washington, DC, USA. IEEE Computer Society.

Sun, A., Lim, E.-P., and Ng, W.-K. (2003). Performance measurement framework for hierarchical text classification. *Journal of the American Society for Information Science and Technology*, 54(11):1014–1028.

Sun, A., Lim, E.-P., Ng, W.-K., and Srivastava, J. (2004). Blocking reduction strategies in hierarchical text classification. *IEEE Trans. on Knowl. and Data Eng.*, 16(10):1305–1308.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 990–998, New York, NY, USA. ACM.

Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Mach. Learn.*, 54(1):45–66.

Tsatsaronis, G., Varlamis, I., and Vazirgiannis, M. (2010). Text relatedness based on a word thesaurus. *J. Artif. Int. Res.*, 37(1):1–40.

Wang, P. and Domeniconi, C. (2008). Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 713–721, New York, NY, USA. ACM.

Wu, F., Zhang, J., and Honavar, V. (2005). *Learning Classifiers Using Hierarchically Structured Class Taxonomies*, pages 313–320. Springer Berlin Heidelberg, Berlin, Heidelberg.

Xiao, Y., Wang, H., and Xu, W. (2015). Parameter selection of gaussian kernel for one-class svm. *IEEE Transactions on Cybernetics*, 45(5):941–953.

Yan, P. and Jin, W. (2012). Improving cross-document knowledge discovery using explicit semantic analysis. In *Data Warehousing and Knowledge Discovery*, pages 378–389. Springer Science Business Media.

Yan, P. and Jin, W. (2016). Building semantic kernels for cross-document knowledge discovery using wikipedia. *Knowledge and Information Systems*, pages 1–24.

Yang, L., Qiu, M., Gottipati, S., Zhu, F., Jiang, J., Sun, H., and Chen, Z. (2013). Cqarank: Jointly model topics and expertise in community question answering. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 99–108, New York, NY, USA. ACM.

Yun, J., Jing, L., Yu, J., and Huang, H. (2012). A multi-layer text classification framework based on two-level representation model. *Expert Systems with Applications*, 39(2):2035 – 2046.

Appendices

Appendix A

Data

The data is comprised of titles and abstracts from articles of the journals. These journals are selected by verifying a journal whose aims and scope matches the keywords defined by NSERC. Each of these journals are verified by confirming the topics discussed in all the articles using tools such as Novanet and Microsoft Academic Search.

A.1 Verifying A Journal

In Section 4.1 and Algorithm 1, we discussed about the using aims and scope of a journal and matching it with keywords of each research topic to find candidate journals for a research topic. Even though a journal may be the most eligible candidate for a research topic, an additional verification is need to ascertain that they belong to a particular research topic than any other. This journal verification can be done using facet tool available at Dalhousie Libraries[1] (via Novanet, Inc) and Microsoft Academic Search[2]. For a given name of the journal and ISSN to Microsoft Academic Search and Novanet, it returns the list of articles published under that journal. Both of these search engines returns facet information to filter the results by year range, authors, publications, journals, and by subjects. We used a journal, European Journal of Immunology, as an example, and the list of subjects returned by Microsoft Academic Search (refer left-most Table in the Figure A.1) and Novanet, Inc (left-most Table in A.2) to verify each selected journal.

The second step in verifying the journals are by matching keywords from Microsoft Academic Search and Novanet is to inspect subjects/topics that irrelevant to keywords for research topic in NSERC. In this case we have identified the close

[1]http://dal.novanet.ca/primo_library/libweb/action/dlSearch.do?institution=DAL&vid=DAL
[2]http://academic.research.microsoft.com/

Research Topic: LSA01
Host-cell interactions; immune response; antigens; antibodies; host-pathogen interactions; immunogenetics; innate immunity; cytokines and antimicrobials; antigen presentation; inflammation; lymphocyte; neutrophil; monocyte; macrophage; sinus; thymus epithelium; lymph node; spleen; chemokine; interleukin; dendritic cell; B cell; T cell; plasma cell; mucosal immunity; immunoglobin; ecological immunology; Toll-like receptors; evolution of immune responses

Table A.1: Keywords for the research topic LSA01 defined by NSERC

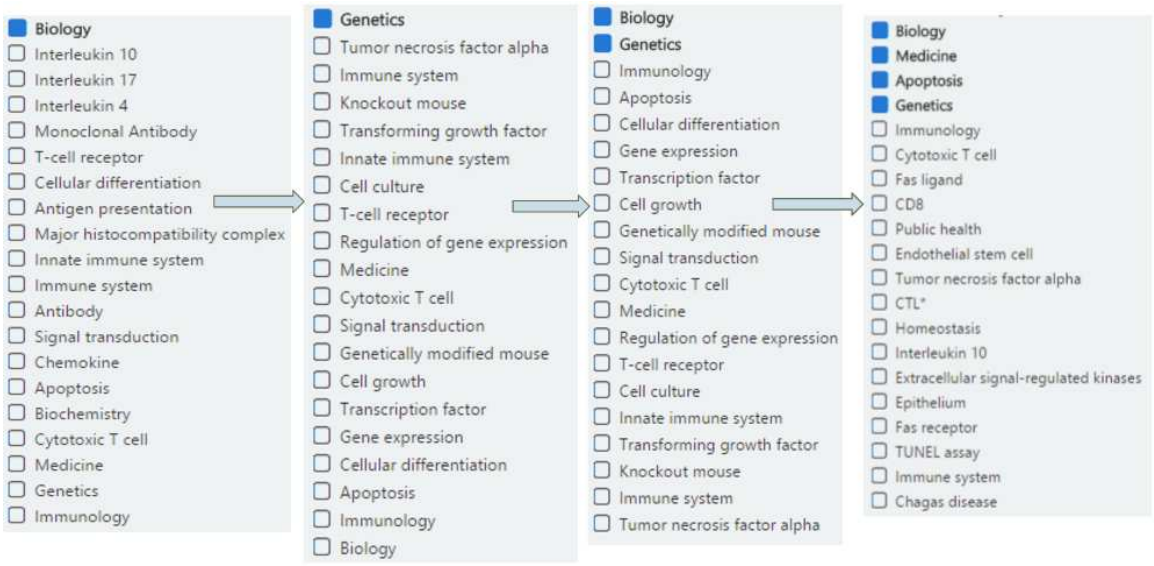


Figure A.1: The articles filtered by subjects to verify journal (European journal of immunology) and inspect irrelevant keywords using Microsoft Academic Search

match of research topic as LSA01 (refer Table A.1). The flow from left to right in Figure A.1 and A.2 shows filtering irrelevant keywords and inspecting the list of subjects/topics returned from these tools. The results shows that the journals do have subject/topics that belong to other research topic according to NSERC, but those articles also have keywords that belong to the research topic of interest and such journals can be mapped to a research topic. It is also important to highlight that each journals were chosen in such a way that they did not get categorized into another research topic as much as possible.

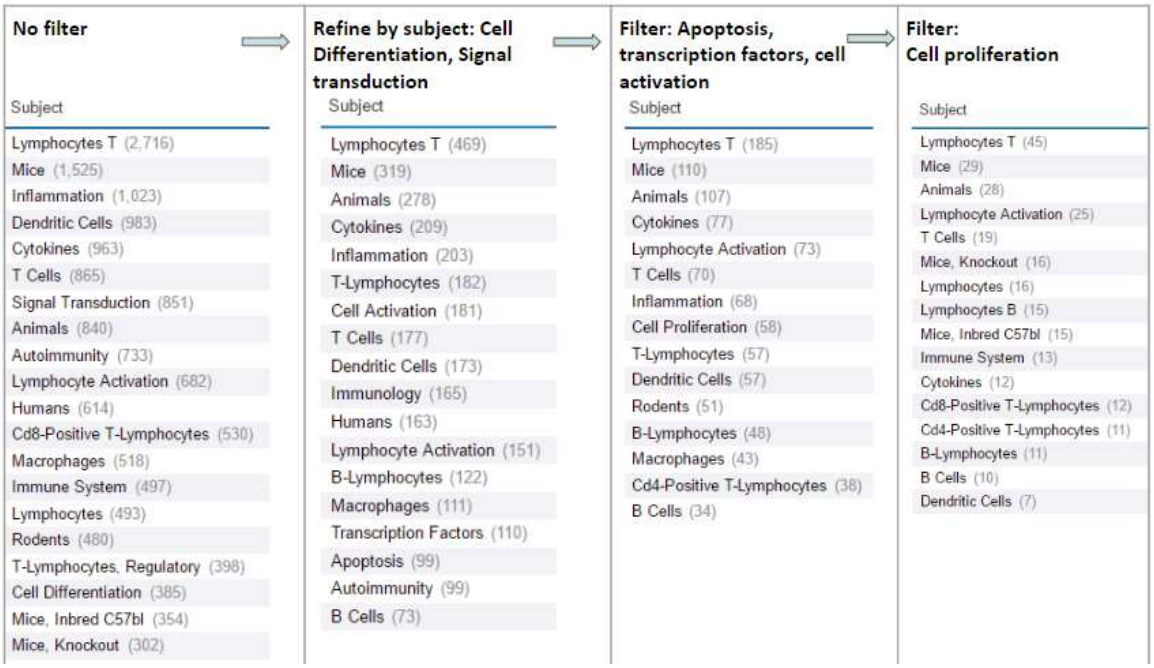


Figure A.2: The articles filtered by subjects to verify journal (European journal of immunology) and inspect irrelevant keywords using Novanet Inc service.

A.2 Journals

This subsection of the thesis, list the journals used to extract articles for each research topic. It shows the list of verified journal that describe the research topic. The ISSN number of each of these journals is used to retrieve articles of journals using Representational State Transfer (REST) API provided by Exlibris Inc through Novanet Inc. The ISSN numbers are not listed along with journals which are available online. The list of journals for each research topic, except LSA03, CS02, CS10, CS13 and CS16, can be found in Tables A.2, A.3, A.4 and A.5.

Label	Research Topic	Journals
LSA01	Immunology	Nature Reviews Immunology; European Journal of Immunology; Annual Review of Immunology; Advances in immunology; Trends in Immunology; Immunological reviews
LSA02	Microbiology	Annual Review of Microbiology; Cellular Microbiology; Journal of Bacteriology; PLOS Pathogens
LSA04	Cellular and Molecular Neuroscience	Cellular and Molecular Neurobiology; Molecular and Cellular NeuroScience; BMC Neuroscience; Molecular Neurodegeneration; Journal of Neuroscience Research; Molecular Brain; Nature Reviews Neurology/Neurobiology
LSA05	Molecular Genetic	Human Molecular Genetics; Human Genetics; Genomic Research; DNA Research
LSA06	Evolutionary and Developmental Genetics	Developmental Cell; Development Genes and Evolution; Annual Review of Cell and Developmental Biology; Stem Cells; Stem Cell Research
LSA07	Cell Signals and Electrical properties	Journal of Signal Transduction; Signal Transduction; Journal of Receptors and Signal Transduction; Cellular Signalling; Cell Communication and Signaling
LSA08	Quantitative Approaches	Bioinformatics - by Oxford; BMC Bioinformatics; NeuroInformatics; Journal of Computational Neuroscience; Journal of Mathematical Biology; Mathematical Biosciences
LSA09	Biochemistry	Annual Review of Biochemistry; Trends in Biochemical Sciences; ACS Chemical Biology; Biochemistry and Cell Biology; Journal of Structural Biology

Table A.2: Selected Journals for an Evaluation Group LSA

A.3 Data Size

For each evaluation group a set of journals are identified which are further verified

Label	Research Topic	Journals
LSB01	Plant Biology	Annual Reviews of Plant Biology; Trends in Plant Science; The Plant Cell; Current Opinion in Plant Biology; Plant Physiology; The Plant Journal; Molecular Plant
LSB02	Food Science	Annual Review of Food Science and Technology; Trends in Food Science & Technology; Critical Reviews in Food Science and Nutrition; Comprehensive Reviews in Food Science and Food Safety; Food Microbiology; Food Chemistry; International Journal of Food Microbiology
LSB03	Physiology and biomechanics (Sports Science)	Journal of Biomechanics; Clinical Biomechanics
LSB06	Behavioral Neuroscience	Neuroscience & Biobehavioral Reviews; Cognitive, Affective, & Behavioral Neuroscience; Behavioral Neuroscience
LSB07	Cognitive Science	Topics in Cognitive Science; Trends in Cognitive Sciences; Journal of Cognitive Neuroscience
LSB09	Nutritional Sciences	Annual Review of Nutrition; American Journal of Clinical Nutrition; Journal of the Academy of Nutrition and Dietetics

Table A.3: Selected Journals for an Evaluation Group LSB

Label	Research Topics	Journals
CS01	Web-Enabled Applications and Services	International Journal of E-Health and Medical Communications; Telemedicine and e-Health; Journal of Medical Internet Research; International Journal of E-Business Research; International Journal of Electronic Business; International Journal of Electronic Business Management; Electronic Government- an International Journal; International Journal of Electronic Governance; Internet and Higher Education; International Journal on E-Learning; The Electronic Journal of e-Learning; European Journal of Open, Distance and e-Learning; E-learning and Education; Electronic Commerce Research and Applications; International Journal of Electronic Commerce; Electronic Commerce Research; e-Service Journal; International Journal of E-Services and Mobile Applications
CS03	Mathematical Computing	Journal of Mathematical Computing; Mathematical Programming; Journal of Computational Mathematics; Foundations of Computational Mathematics; Applied Mathematics and Computation; Mathematical and Computer Modelling; Discrete Optimization
CS04	Theory of Computing	Computational Complexity; ACM Transactions on Computational Logic; Journal of Complexity
CS05	Algorithms and Data Structures	Journal of Discrete Algorithms; Journal of Graph Algorithms and Applications; Algorithms for Molecular Biology; ACM Transactions on Algorithms; Journal of Experimental Algorithmics; Algorithmica; Combinatorica; Journal of Combinatorial Optimization
CS06	Computer Networks	Computer Communications; ACM Transactions on Sensor Networks; Ad Hoc Networks; Peer-to-Peer Networking and Applications; IEEE Network
CS07	Quantum Computing	International Journal of Quantum Information; Quantum Information Processing
CS08	Information Systems	Information Systems; ACM Transactions on Information Systems; International Journal of Geographical Information Science; Decision Support Systems; International Journal of Medical Informatics
CS09	Security and Privacy	International Journal of Information Security; Computers & Security; Designs, Codes and Cryptography
CS11	Programming Languages	ACM Transactions on Programming Languages and Systems; Journal of Functional Programming; Systems and Structures

Table A.4: Selected Journals for an Evaluation Group CS - Part 1

Label	Research Topic	Journals
CS12	Software Engineering	Automated Software Engineering; IEEE Transactions on Software Engineering; ACM Transactions on Software Engineering and Methodology; Empirical Software Engineering; Requirements Engineering; Software Testing, Verification and Reliability; Software Quality Journal; International Journal of High Performance Computing Applications
CS14	Computing Systems	International Journal of Embedded and Real-Time Communication Systems; ACM Transactions in Embedded Computing Systems; Operating Systems Review; International Journal of Green Computing
CS15	Parallel and Distributed Computing	Journal of Parallel and Distributed Computing; Parallel Computing; IEEE Transactions on Parallel and Distributed Systems; Cluster Computing; Journal of Grid Computing; Distributed Computing
CS17	Human Computer Interaction	International Journal of Human-Computer Studies; ACM Transactions on Computer-Human Interaction; Interacting with Computers; User Modeling and User-Adapted Interaction; Journal on Multimodal User Interfaces
CS18	AI	Artificial Intelligence; Knowledge-Based Systems; ACM Transactions on Intelligent Systems and Technology; Machine Learning; Swarm Intelligence; Journal of Semantics; IEEE Intelligent Systems; IEEE Transactions on Computational Intelligence and AI in Games
CS19	Computer Graphics and Visualization	ACM Transactions on Graphics; IEEE Transactions on Visualization and Computer Graphics; Computer and Graphics; IEEE Computer Graphics and Applications
CS20	Bioinformatics and Bioinspired Computing	Health Informatics Journal; Algorithms for Molecular Biology; IEEE/ACM Transactions on Computational Biology and Bioinformatics; Journal of Bioinspired Computation; BMC Bioinformatics
CS21	Computer Vision and Robotics	Computer Vision and Image Understanding; Image and Vision Computing; International Journal of Computer Vision; The International Journal of Robotics Research; International Journal of Advanced Robotic Systems

Table A.5: Selected Journals for an Evaluation Group CS - Part 2

Label	Research Topic	# Journals	Total Docs
LSA01	Immunology	6	9467
LSA02	Microbiology	4	11061
LSA04	Cellular and Molecular Neuroscience	4	11548
LSA05	Molecular Genetic	4	10824
LSA06	Evolutionary and Developmental Genetics	5	8793
LSA07	Cell Signals and Electrical properties	6	4366
LSA08	Quantitative Approaches	6	21438
LSA09	Biochemistry	5	3424
Total		**40**	**80,921**

Table A.6: Number of articles retrieved for each research topic of evaluation group LSA and in total summary.

Labels	Research Topic	# Journals	Total Docs
LSB01	Plant Biology	4	6940
LSB02	Food Science	5	10250
LSB03	Physiology and Biomechanics	2	4618
LSB06	Behavioural Neuroscience	3	4676
LSB07	Cognitive Science	3	3483
LSB09	Nutritional Sciences	3	3667
Total		**20**	**35,751**

Table A.7: Number of articles retrieved for each research topic of evaluation group LSB and in total summary.

Labels	Research Topic	#	Total Docs
CS01	Web-Enabled Applications and Services	20	5232
CS03	Mathematical Computing	4	3247
CS04	Theory of Computing	2	1149
CS05	Algorithms and Data Structures	8	4306
CS06	Computer Networks	6	6040
CS07	Quantum Computing	2	2283
CS08	Information Systems	6	4471
CS09	Security and Privacy	3	3085
CS11	Programming Languages	3	829
CS12	Software Engineering	8	3169
CS14	Computing Systems	3	932
CS15	Parallel and Distributed Computing	6	5879
CS17	Human Computer Interaction	5	2323
CS18	AI	7	5664
CS19	Computer Graphics and Visualization	4	4937
CS20	Bioinformatics and Bioinspired Computing	4	1635
CS21	Computer Vision and Robotics	5	6750
Total		**96**	**61,931**

Table A.8: Number of articles retrieved for each research topic of evaluation group CS and in total summary.

Appendix B

O-vs-R Classifiers

Classifier	Accuracy	Precision	Recall	F1-score
Ridge Classifier	81.7	81	82	81
Perceptron	77.3	78	77	77
Multinomial NB	74.2	76	74	71
Bernoulli NB	78.4	81	78	79
Linear SVC	83.5	83	83	83

Table B.1: Performance on top 500 features selected using χ^2 on LSA

Classifier	Accuracy	Precision	Recall	F1-score
Ridge Classifier	88.3	88.1	88.2	88.0
Perceptron	84.7	84.5	84.5	84.5
Multinomial NB	83.7	83.5	83.7	83.5
Bernoulli NB	82.5	83.9	82.5	82.9
Linear SVC	89.2	89.1	89.2	89.1

Table B.2: Performance on top 10,000 features selected using χ^2 on LSA

Research Topics	BoW	BoC	BoK
	F1-Score	F1-Score	F1-Score
LSA01	0.8971	0.8585	0.8510
LSA02	0.9079	0.8431	0.8454
LSA04	0.9118	0.8267	0.8256
LSA05	0.8623	0.7665	0.7501
LSA06	0.8747	0.7621	0.7373
LSA07	0.7368	0.6921	0.6753
LSA08	0.9664	0.9043	0.8890
LSA09	0.6998	0.5044	0.4203

Table B.3: One-vs-Rest classification on evaluation group LSA for different document representation

Research Topics	BoW	BoC	BoK
	F1-Score	F1-Score	F1-Score
LSB01	0.9846	0.9631	0.9606
LSB02	0.9761	0.9261	0.9198
LSB03	0.9945	0.9586	0.9528
LSB06	0.8687	0.8023	0.8
LSB07	0.8319	0.7656	0.7775
LSB09	0.9567	0.8246	0.8268

Table B.4: One-vs-Rest classification on evaluation group LSB for different document representation

Research Topic	BoW	BoC	BoK
	F1-Score	F1-Score	F1-Score
CS01	0.7905	0.6634	0.6591
CS03	0.7951	0.6881	0.6844
CS04	0.7126	0.4257	0.4455
CS05	0.7311	0.6376	0.6525
CS06	0.7739	0.7086	0.7119
CS07	0.9723	0.8556	0.8594
CS08	0.6420	0.4623	0.4260
CS09	0.7492	0.5558	0.5698
CS11	0.6595	0.5594	0.5411
CS12	0.7703	0.5854	0.5789
CS14	0.4818	0.3387	0.3559
CS15	0.7072	0.5302	0.5516
CS17	0.6438	0.4578	0.4670
CS18	0.7174	0.5248	0.5406
CS19	0.8372	0.6011	0.6016
CS20	0.7591	0.6093	0.6278
CS21	0.8739	0.6602	0.6532

Table B.5: Performance of One-vs-Rest Classifier on CS and Different Representations (10-fold CV)

Appendix C

Hierarchical Classifier

C.1 Comparison of flat and hierarchical classifier

Research Topics	BoW (Flat classification)			BoW (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSA01	0.8839	0.9002	0.892	0.88	0.89	0.89
LSA02	0.8788	0.9229	0.9004	0.86	0.91	0.88
LSA04	0.8814	0.8677	0.8745	0.85	0.85	0.85
LSA05	0.8547	0.8563	0.8555	0.84	0.85	0.84
LSA06	0.8565	0.8487	0.8526	0.84	0.84	0.84
LSA07	0.7808	0.758	0.7692	0.76	0.76	0.76
LSA08	0.8912	0.9403	0.9151	0.89	0.9	0.89
LSA09	0.7699	0.5994	0.674	0.71	0.59	0.65

Table C.1: Comparison of Flat and Hierarchical Classifier on BOW of LSA data

Research Topics	BoW (Flat classification)			BoW (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
CS01	0.7752	0.8113	0.7928	0.76	0.8	0.78
CS03	0.7931	0.7793	0.7862	0.76	0.79	0.77
CS04	0.723	0.6091	0.6612	0.7	0.59	0.64
CS05	0.7677	0.7467	0.7571	0.76	0.75	0.75
CS06	0.7601	0.8309	0.7939	0.75	0.82	0.79
CS07	0.9481	0.9742	0.961	0.95	0.97	0.96
CS08	0.6736	0.6006	0.635	0.64	0.58	0.61
CS09	0.7007	0.744	0.7217	0.71	0.75	0.73
CS11	0.6777	0.6245	0.65	0.68	0.66	0.67
CS12	0.7397	0.7154	0.7274	0.72	0.71	0.72
CS14	0.6385	0.3168	0.4235	0.6	0.31	0.41
CS15	0.7013	0.7013	0.7013	0.68	0.7	0.69
CS17	0.6903	0.631	0.6593	0.68	0.63	0.65
CS18	0.6841	0.6728	0.6784	0.66	0.68	0.67
CS19	0.8212	0.829	0.8251	0.79	0.83	0.81
CS20	0.5329	0.1874	0.2773	0.38	0.27	0.31
CS21	0.8314	0.8864	0.858	0.8	0.89	0.84

Table C.2: Comparison of Flat and Hierarchical Classifier on BOW of CS data

Research Topics	BoW (Flat classification)			BoW (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSB01	0.8472	0.8502	0.8487	0.9	0.84	0.87
LSB02	0.8659	0.8933	0.8794	0.96	0.94	0.95
LSB03	0.86	0.8696	0.8648	0.95	0.9	0.93
LSB06	0.6699	0.6537	0.6617	0.76	0.74	0.75
LSB07	0.6329	0.6972	0.6635	0.74	0.74	0.74
LSB09	0.7774	0.7715	0.7744	0.89	0.86	0.87

Table C.3: Comparison of Flat and Hierarchical Classifier on BOW of LSB data

Research Topics	BoC (Flat classification)			BoC (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSA01	0.8168	0.8549	0.8354	0.82	0.84	0.83
LSA02	0.8103	0.8392	0.8245	0.78	0.83	0.81
LSA04	0.7707	0.7868	0.7787	0.74	0.77	0.76
LSA05	0.7605	0.7265	0.7431	0.74	0.73	0.73
LSA06	0.763	0.7428	0.7528	0.73	0.74	0.74
LSA07	0.6659	0.673	0.6695	0.65	0.66	0.66
LSA08	0.7042	0.8175	0.7567	0.74	0.73	0.73
LSA09	0.5063	0.3553	0.4176	0.47	0.37	0.42

Table C.4: Comparison of Flat and Hierarchical Classifier on BOc of LSA data

Research Topics	BoW (Flat classification)			BoW (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
CS01	0.7752	0.8113	0.7928	0.76	0.8	0.78
CS03	0.7931	0.7793	0.7862	0.76	0.79	0.77
CS04	0.723	0.6091	0.6612	0.7	0.59	0.64
CS05	0.7677	0.7467	0.7571	0.76	0.75	0.75
CS06	0.7601	0.8309	0.7939	0.75	0.82	0.79
CS07	0.9481	0.9742	0.961	0.95	0.97	0.96
CS08	0.6736	0.6006	0.635	0.64	0.58	0.61
CS09	0.7007	0.744	0.7217	0.71	0.75	0.73
CS11	0.6777	0.6245	0.65	0.68	0.66	0.67
CS12	0.7397	0.7154	0.7274	0.72	0.71	0.72
CS14	0.6385	0.3168	0.4235	0.6	0.31	0.41
CS15	0.7013	0.7013	0.7013	0.68	0.7	0.69
CS17	0.6903	0.631	0.6593	0.68	0.63	0.65
CS18	0.6841	0.6728	0.6784	0.66	0.68	0.67
CS19	0.8212	0.829	0.8251	0.79	0.83	0.81
CS20	0.5329	0.1874	0.2773	0.38	0.27	0.31
CS21	0.8314	0.8864	0.858	0.8	0.89	0.84

Table C.5: Comparison of Flat and Hierarchical Classifier on BOc of CS data

Research Topics	BoC (Flat classification)			BoC (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSB01	0.8472	0.8502	0.8487	0.85	0.74	0.79
LSB02	0.8659	0.8933	0.8794	0.87	0.83	0.85
LSB03	0.86	0.8696	0.8648	0.88	0.79	0.83
LSB06	0.6699	0.6537	0.6617	0.64	0.58	0.61
LSB07	0.6329	0.6972	0.6635	0.63	0.62	0.62
LSB09	0.7774	0.7715	0.7744	0.77	0.71	0.74

Table C.6: Comparison of Flat and Hierarchical Classifier on BOc of LSB data

Research Topics	BoK (Flat classification)			BoK (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSA01	0.8146	0.8553	0.8345	0.83	0.85	0.84
LSA02	0.8021	0.8365	0.8189	0.77	0.84	0.8
LSA04	0.7508	0.7853	0.7677	0.72	0.77	0.74
LSA05	0.7502	0.7169	0.7332	0.73	0.71	0.72
LSA06	0.7399	0.7195	0.7296	0.72	0.71	0.71
LSA07	0.6604	0.6447	0.6525	0.64	0.63	0.64
LSA08	0.687	0.8077	0.7425	0.73	0.71	0.72
LSA09	0.5216	0.3091	0.3881	0.48	0.32	0.39

Table C.7: Comparison of Flat and Hierarchical Classifier on BOK of LSA data

Research Topics	BoK (Flat classification)			BoK (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
CS01	0.5949	0.6468	0.6198	0.56	0.63	0.59
CS03	0.6192	0.6576	0.6378	0.57	0.65	0.61
CS04	0.5412	0.4182	0.4718	0.52	0.42	0.46
CS05	0.5635	0.6412	0.5998	0.56	0.65	0.6
CS06	0.64	0.7457	0.6888	0.63	0.75	0.68
CS07	0.8595	0.8304	0.8447	0.84	0.83	0.83
CS08	0.4529	0.384	0.4156	0.41	0.38	0.4
CS09	0.5503	0.517	0.5332	0.54	0.5	0.52
CS11	0.5822	0.5415	0.5611	0.6	0.55	0.57
CS12	0.5536	0.4977	0.5242	0.51	0.5	0.5
CS14	0.5424	0.2443	0.3368	0.48	0.23	0.31
CS15	0.5767	0.5035	0.5377	0.56	0.5	0.53
CS17	0.4331	0.3313	0.3754	0.36	0.36	0.36
CS18	0.4864	0.4214	0.4516	0.4	0.46	0.43
CS19	0.6128	0.5495	0.5794	0.56	0.56	0.56
CS20	0.1667	0.0063	0.0122	0.16	0.11	0.13
CS21	0.6294	0.6212	0.6252	0.56	0.64	0.6

Table C.8: Comparison of Flat and Hierarchical Classifier on BOK of CS data

Research Topics	BoK (Flat classification)			BoK (Hierarchical Classification)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSB01	0.8456	0.864	0.8547	0.84	0.74	0.79
LSB02	0.8579	0.8822	0.8699	0.85	0.82	0.84
LSB03	0.8501	0.8703	0.8601	0.87	0.8	0.83
LSB06	0.6743	0.6485	0.6612	0.63	0.58	0.6
LSB07	0.6194	0.7002	0.6573	0.62	0.62	0.62
LSB09	0.7567	0.7696	0.7631	0.73	0.71	0.72

Table C.9: Comparison of Flat and Hierarchical Classifier on BOK of LSB data

Appendix D

t-SNE

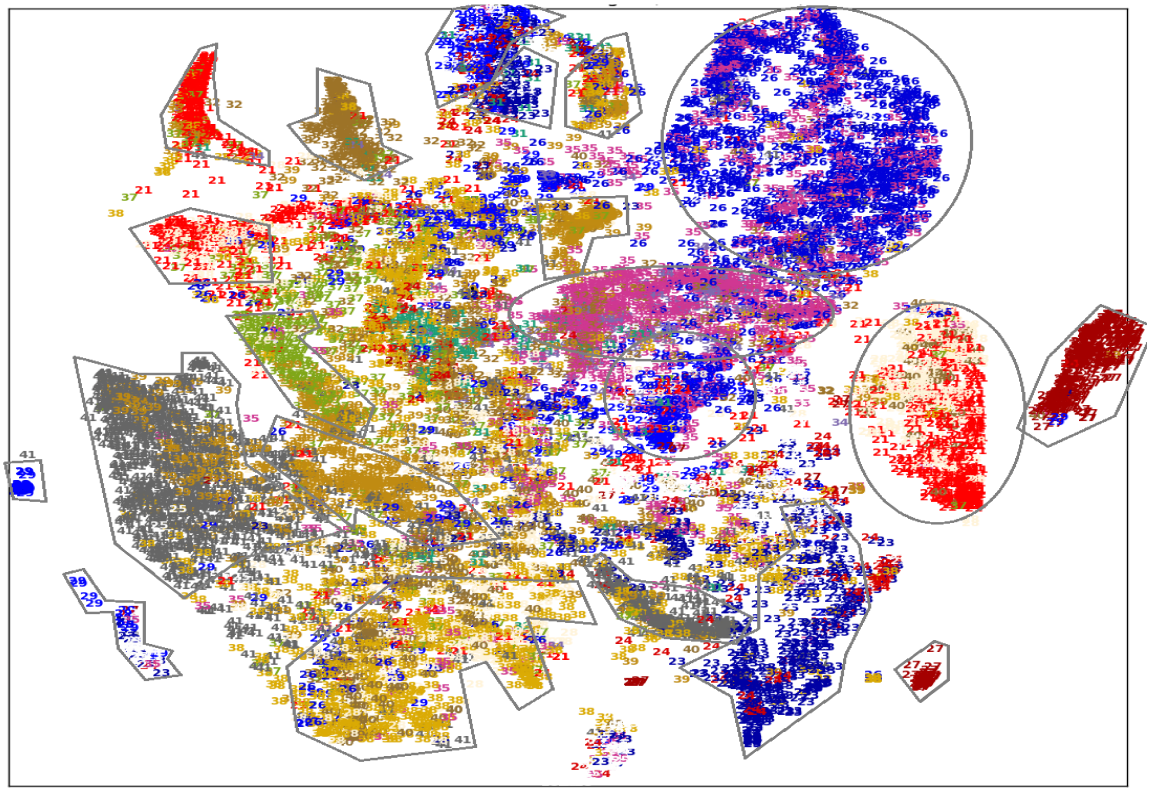


Figure D.1: 2D plot of 20% samples from evaluation group CS on BOW using t-SNE.

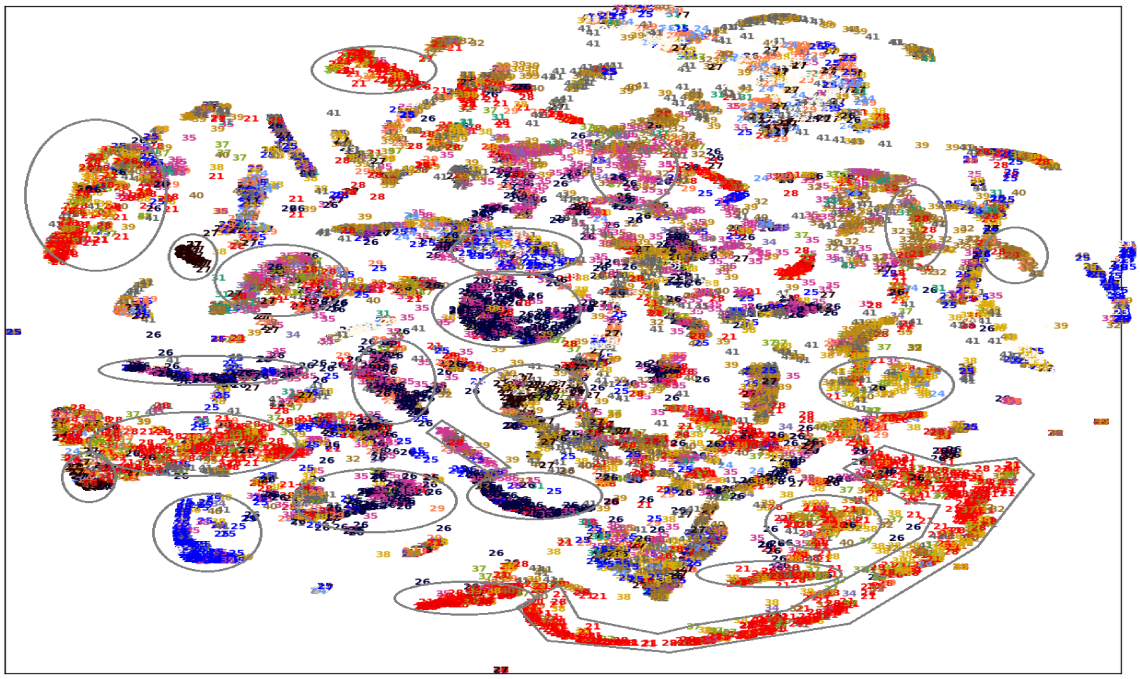


Figure D.2: 2D plot of 20% samples from evaluation group CS on BOC using t-SNE.

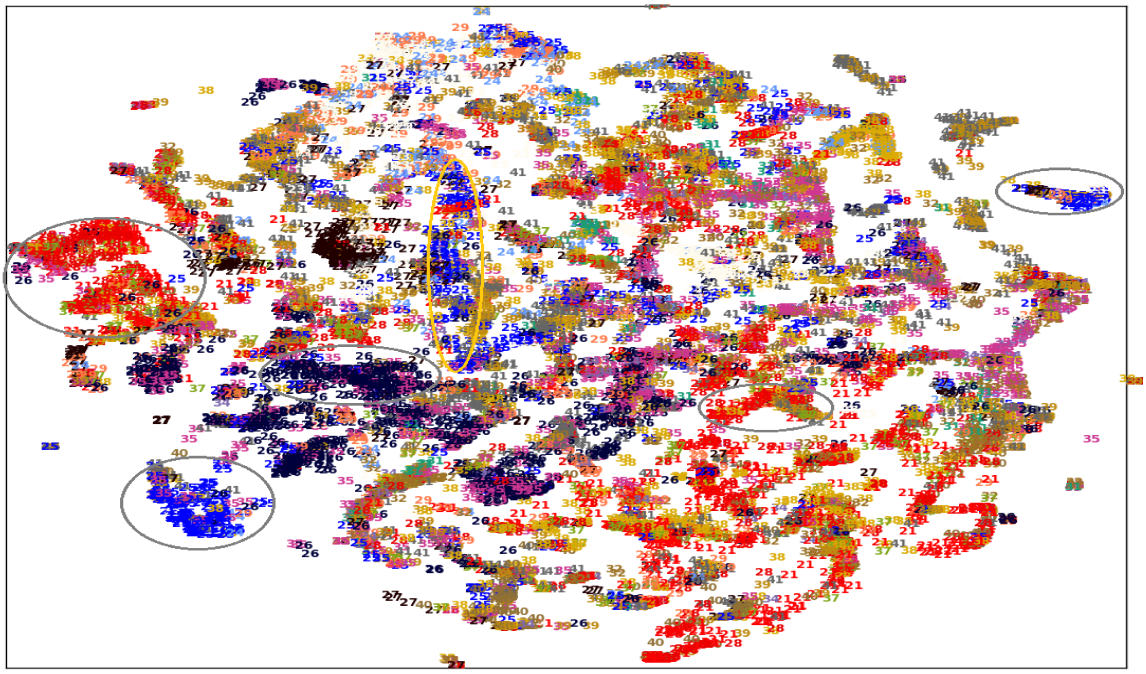


Figure D.3: 2D plot of 20% samples from evaluation group CS on BOK using t-SNE.

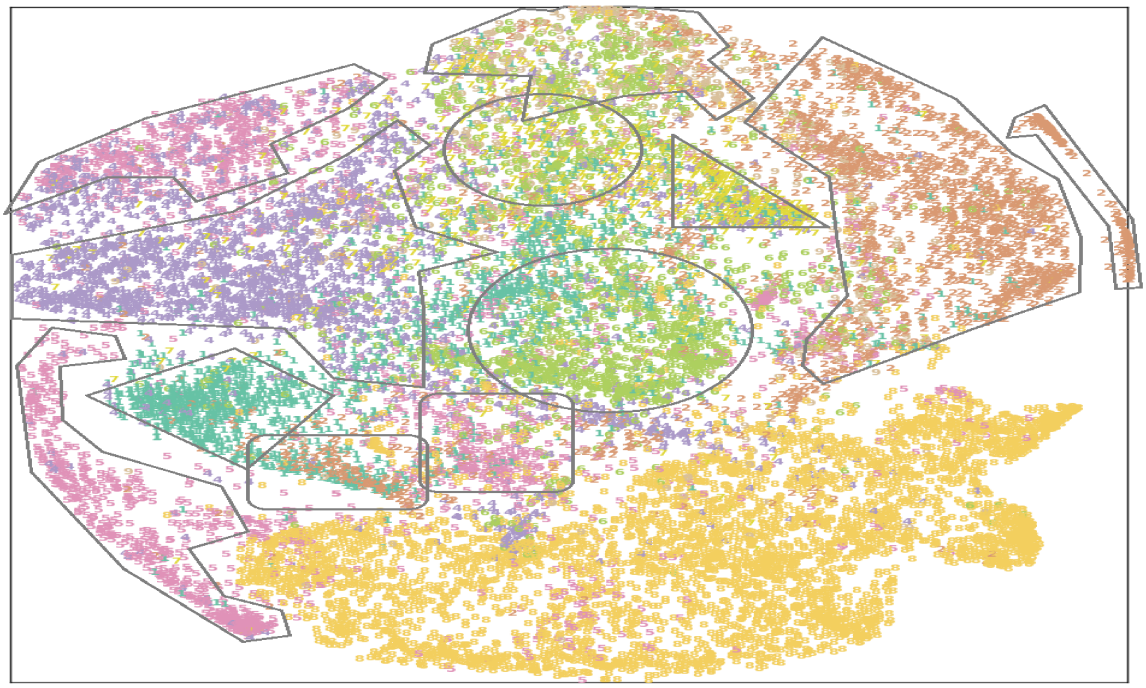


Figure D.4: 2D plot of 20% samples from evaluation group LSA on BOW using t-SNE.

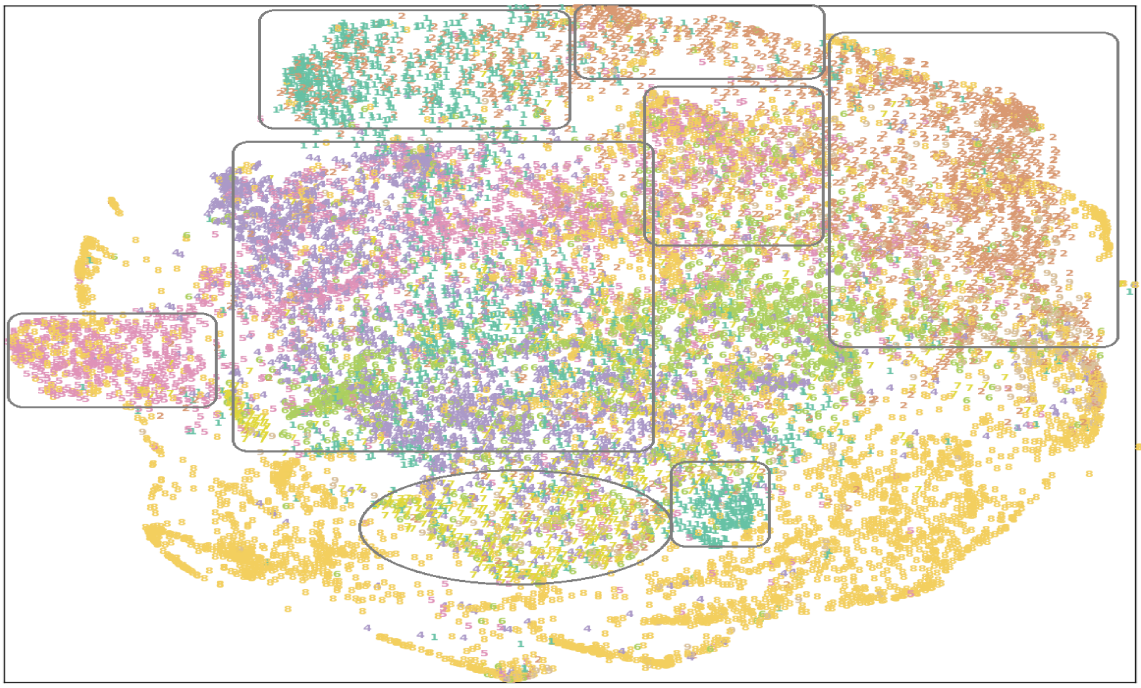


Figure D.5: 2D plot of 20% samples from evaluation group LSA on BOC using t-SNE.

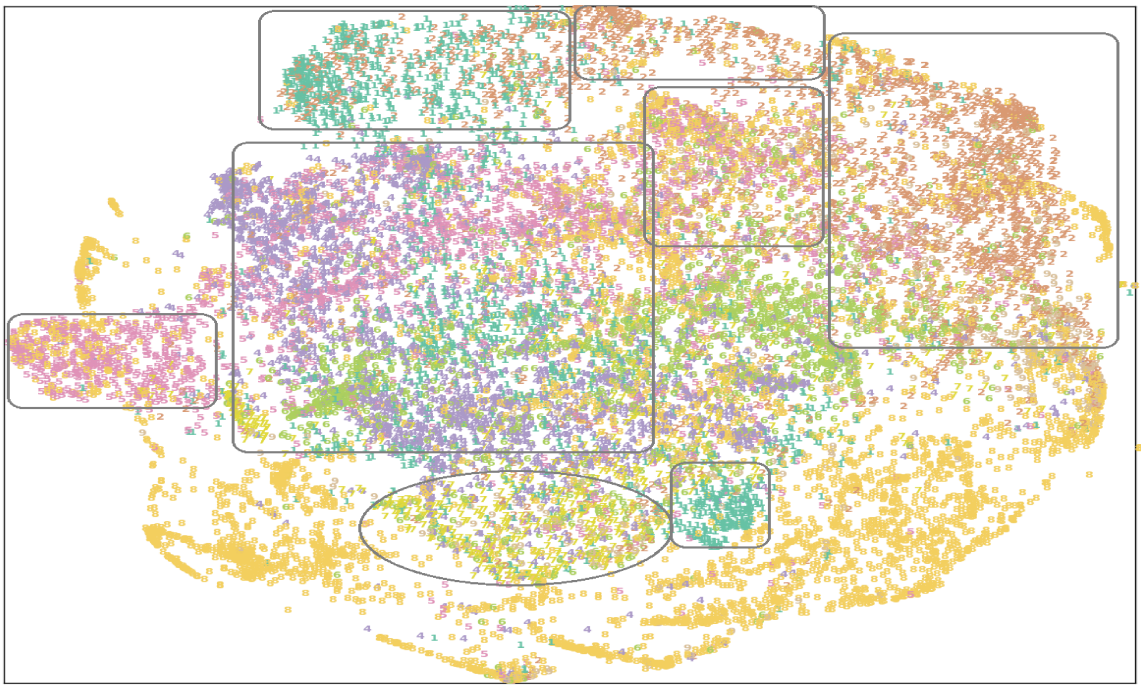


Figure D.6: 2D plot of 20% samples from evaluation group LSA on BOK using t-SNE.

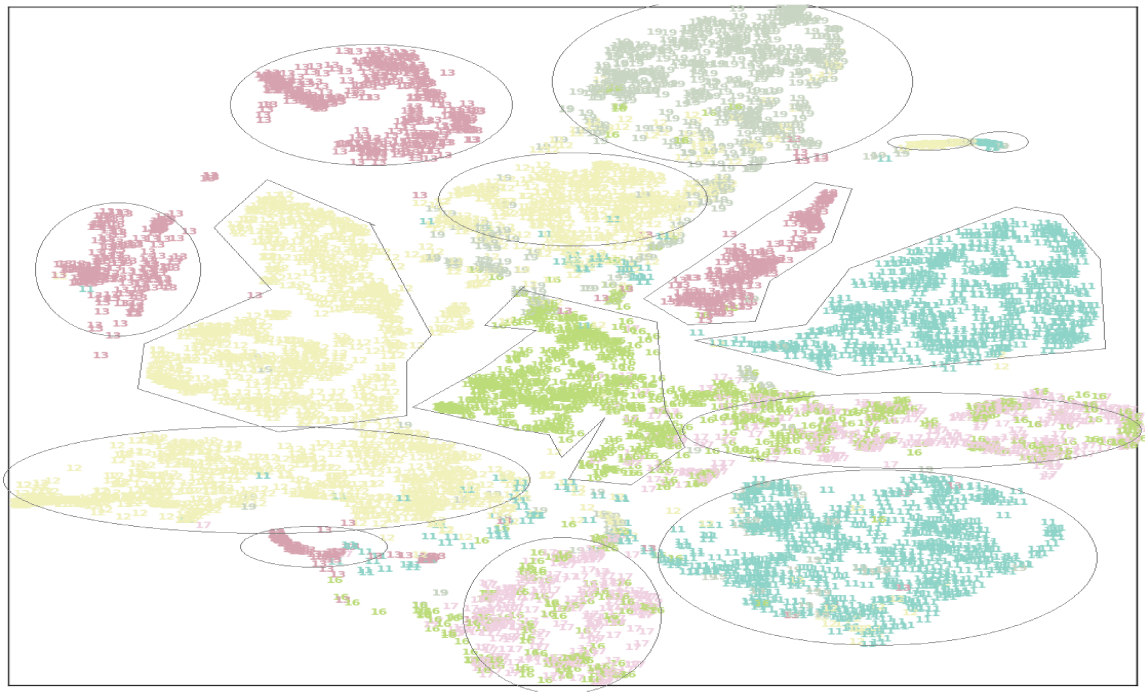


Figure D.7: 2D plot of 20% samples from evaluation group LSB on BOW using t-SNE.

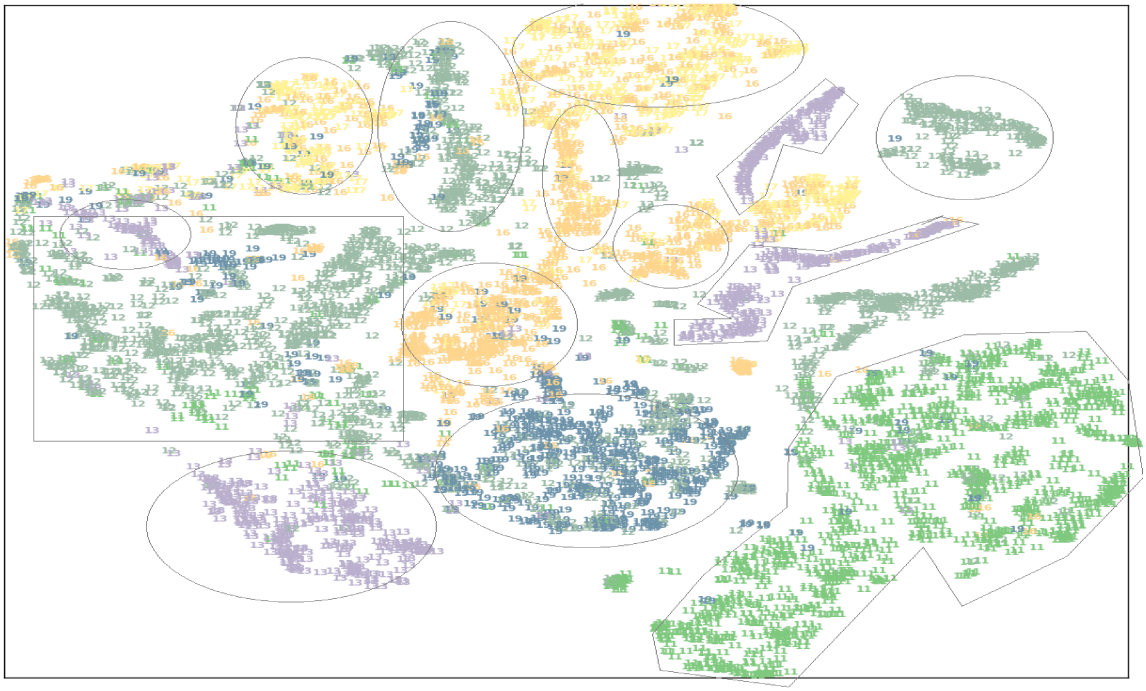


Figure D.8: 2D plot of 20% samples from evaluation group LSB on BOC using t-SNE.

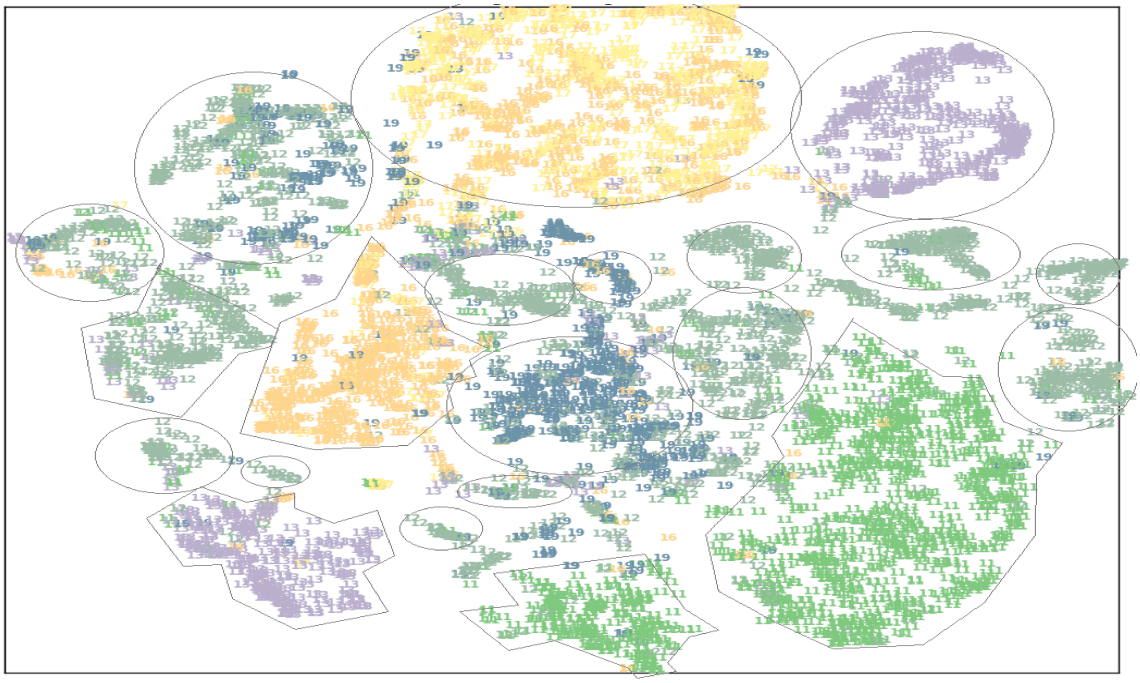


Figure D.9: 2D plot of 20% samples from evaluation group LSB on BOK using t-SNE.

Appendix E

One-Class SVM

E.1 Performance of One-Class SVM on 10-Fold CV

Figure E.1: 10-Fold CV of One-Class classifier for evaluation group LSA02 on various 'nu' parameter.

Figure E.2: 10-Fold CV of One-Class classifier for evaluation group LSA03 on various 'nu' parameter.

E.2 Analysis Of Documents For Multi-Labeling

Id: 2334: *The immune system can be roughly divided into innate and adaptive compartments. The adaptive compartment includes the B and T lymphocytes, whose antigen receptors are generated by recombination of gene segments. The consequence is that the creation of self-reactive lymphocytes is unavoidable. For the host to remain viable, the immune system has evolved a strategy for removing autoimmune lymphocytes during development. This review discusses how T lymphocytes are generated, how they recognize antigens, and how their antigen receptor directs the removal of self-reactive T cells.*

The article (Id: 2334) in above text box is published in year 2005 in journal: Journal of Receptors and Signal Transduction. Reading the abstract of this article it should be classified into LSA01 (Immunology), but last line talks about receptor of antigen which is from research topic LSA07 (Cell Signals and Electrical Properties). So this article belongs to both LSA01 and LSA07.

Appendix F

Sunflower

Sunflower is an extension of Tulip (Lipczak et al., 2014) which uses 120 different languages to interpret accurate categories of a concept. Sunflower tool has an API to extract categories from Wikipedia. The categories from this tool has a tree structure which can be pruned using depth and width parameters. Since pruning is important to avoid too general or irrelevant categories, Table F.1 shows parameter optimization for extracting the categories. Though the performance of hierarchical classifier on width and depth greater than 2 is better, but it include more noise or irrelevant categories for a concept.

Avg	Sunflower Configuration	Accuracy	Precision	Recall	F1-Score
Weighted	Depth=1, Width=1	0.6157	0.6251	0.6244	0.6207
Macro	Depth=1, Width=1		0.5491	0.5282	0.5334
Weighted	Depth=2,Width=2	0.6613	0.6715	0.6785	0.6544
Macro	Depth=2,Width=2		0.6091	0.5944	0.5983
Weighted	Depth=3, Width=3	0.6637	0.6889	0.6841	0.6812
Macro	Depth=3, Width=3		0.6119	0.5917	0.6016
Weighted	Depth=4,Width=4	0.676	0.69	0.6831	0.6805
Macro	Depth=4,Width=4		0.6191	0.5944	0.6064

Table F.1: Optimizing depth and width of categories tree from Sunflower.