

# Comparison of Active Learning Based Hierarchical Classification Approaches on Twitter

by

Rashid Zaman

Submitted to the  
Graduate School of Engineering and Natural Sciences  
in partial fulfillment of the requirements for the degree of  
Master of Science

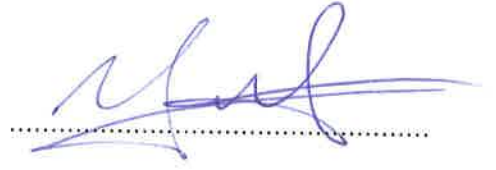
SABANCI UNIVERSITY

December 2015

Comparison of Active Learning Based Hierarchical Classification Approaches on  
Twitter

Approved by:

Prof. Dr. Yücel Saygın  
(Thesis Supervisor)



Assoc. Prof. Dr. Hakan Erdoğan



Assoc. Prof. Dr. Şule Gündüz Ögüdücü



Date of Approval: ...30-12-2015.

©Rashid Zaman 2015  
All Rights Reserved

*To all those who live for others...*

## Acknowledgements

I would like to thank my thesis advisor **Dr. Yücel Saygın** for his scientific support, encouragement, motivation and appreciation.

I am thankful to my thesis evaluation committee members for their helpful comments.

**Dr. Kamer Kaya** and **Dr. Berin Yanikoğlu** are lauded for their provision of computing facilities.

I extend my kind regards to **Dr. Myra Spiliopoulou**, Head KMD Lab, University of Magdeburg, Germany for providing insight to active learning in text classification.

Credit goes to my colleagues and friends **Stefan Rübiger**, **Arsalan Javeed**, **Zoya Khalid** and especially **Kousar Aslam** for the invaluable friendship, help and guidance during all my studies and research period at Sabanci university.

I am thankful to **my family** for their much-needed love and support.

Last but not the least, I am highly grateful to **Higher Education Commission, Pakistan** for funding my studies at Sabanci University.

**Rashid Zaman**

# Comparison of Active Learning Based Hierarchical Classification Approaches on Twitter

Rashid Zaman

Computer Science and Engineering, Master's Thesis, 2015

Thesis Supervisor: Prof. Dr. Yücel Saygın

## Abstract

Real world data is mostly multi-labeled i.e., it belongs to multiple classes simultaneously, as opposed to single labeled data belonging to a single class. At times these multiple labels fit into a logical hierarchy such that parent labels up in the hierarchy are generic and the related child labels down the hierarchy are more specific. Most of the machine learning classifiers are either serving single label classification tasks or have been transformed to perform flat multi-label classification. At present, dedicated classifiers for hierarchical classification do not exist. For the purpose, strategies are designed relying on the single labeled classifiers to perform hierarchical classification. Four such strategies are well-known in literature. Hierarchical classification has been researched in many domains like text categorization, webpages classification and medical diagnosis and has been found very useful. So far Twitter has been neglected by the researchers in hierarchical classification perspective. For developing supervised models labeled data is needed and labeling task requires resources in terms of humans, money and time, delimiting the amount of data which can be labeled. Active learning, a type of supervised learning, achieves acceptable performance with minimal amount of labeled data as compared to supervised learning models. In active learning, the learner selects the most informative unlabeled instances and is labeled by the experts. This makes possible to achieve comparable model performance to that of supervised learning with lesser labeling effort and resources. Active learning is well-suited to the situations where unlabeled data is abundantly available. Hierarchical classification of tweets complemented by active learning as a viable labeling mechanism presents an interesting research problem. We implemented the prevailing four hierarchical classification approaches with active learning for twitter domain. Based on the results, we can safely say that active learning is equally beneficial in Twitter. Comparing the results of the four approaches, hierarchical prediction through flat classification with active learning approach outperforms the other approaches.

# Twitter Alanında Hiyerarşik Sınıflandırma Yöntemini Temel Alan Aktif Öğrenmenin Karşılaştırılması

Rashid Zaman

Bilgisayar Bilimi & Mühendisliği, Yüksek Lisans Tezi, 2015

Tez Danışmanı: Doç. Dr. Yücel Saygın

## Özet

Gerçek hayatta veriler sıklıkla çok etiketlidir, yani aynı anda birden fazla sınıfa ya da kategoriye ait olabilirler. Bazen bu sınıflar üst seviyeler genel, alt seviyeler ise daha özel olacak şekilde mantıksal bir hiyerarşi oluşturur. Makine öğrenmesi kapsamında geliştirilmiş olan çoğu sınıflandırma yöntemi ya tek etiketli sınıflandırma yapar ya da çok etiketli sınıflandırma yapacak şekilde değiştirilir. Hiyerarşik sınıflandırma yapmak için uygun sınıflandırma yöntemleri henüz bulunmamaktadır ancak bunun için tek etiketli sınıflandırmayı baz alan stratejiler geliştirilmiştir. Bu stratejilerden dördü literatürde iyi bilinmektedir. Hiyerarşik sınıflandırma metin ketogorizasyonu, web sayfası sınıflandırması, medikal tanı gibi alanlarda çalışılmış ve etkinliği gösterilmiştir. Ancak şu ana kadar Twitter'a özel hiyerarşik sınıflandırma üzerine çalışılmamıştır. Bunun yanında, gözetimli öğrenme yöntemleri için etiketli verilere ihtiyaç duyulur ve etiketleme için insan gücü, zaman, ve maddi kaynak gerekir. Bu da etiketlenen verilerin sınırlı olmasına sebep olur ve aktif öğrenme bu anlamda daha az verinin etiketlenmesi ile düzgün modeller oluşturulmasını sağlar. Aktif Öğrenmede, en fazla bilgi içeren etiketlenmemiş veri seçilir ve uzmanlara etiketlemesi için sunulur. Bu sayede gözetimli öğrenmeye yakın bir performansla daha az etiketli veri kullanılarak model oluşturulması sağlanır. Aktif öğrenme, etiketlenmemiş verilerin çok olduğu durumlar için uygundur. Tweetlerin hiyerarşik sınıflandırmasının aktif öğrenme ile gerçekleştirilmesi de bu bakımdan anlamlı bir araştırma alanıdır. Bu tezde, önde gelen 4 hiyerarşik sınıflandırma yaklaşımını uyguladık ve aktif öğrenme için bunları Twitter ortamına uyarladık. Elde ettiğimiz sonuçlar baz alındığında, aktif öğrenmenin Twitter alanında faydalı olduğunu görmekteyiz. Uyguladığımız dört ana yaklaşımı karşılaştırdığımızda düz sınıflandırmalı hiyerarşik kestirim kullanılarak yapılan aktif öğrenmenin diğer üç yöntemden daha iyi sonuçlar verdiğini gördük.

# Contents

Acknowledgements	iv
Abstract	v
Özet	vi
List of Figures	ix
List of Tables	x
Abbreviations	xi
<b>1 Introduction</b>	<b>1</b>
<b>2 Preliminaries and Background</b>	<b>5</b>
2.1 Hierarchical Classification . . . . .	5
2.1.1 Type of Hierarchy . . . . .	7
2.1.2 Depth of Hierarchy . . . . .	8
2.1.3 Classification Approaches . . . . .	8
2.1.4 Applications of Hierarchical Classification . . . . .	12
2.2 Active Learning . . . . .	13
2.2.1 Learner . . . . .	15
2.2.2 Initial Seed . . . . .	15
2.2.3 Sampling Strategies . . . . .	16
2.2.3.1 Synthetic Queries . . . . .	18
2.2.3.2 Batch-mode Queries . . . . .	19
2.2.4 Unlabeled Data (Pool) . . . . .	19
2.2.5 Oracles . . . . .	19
2.2.6 Stoppage Criteria . . . . .	20
2.2.7 Applications of Active Learning . . . . .	21
<b>3 Research Problem</b>	<b>22</b>



---

3.1	Problem Definition and Proposed Approach . . . . .	22
3.2	Research Hypotheses . . . . .	23
3.3	Related Work . . . . .	23
<b>4</b>	<b>Experiments Design</b>	<b>25</b>
4.1	The Hierarchy . . . . .	25
4.2	Experiments . . . . .	26
4.2.1	Big Bang . . . . .	26
4.2.2	Single Flat Classification . . . . .	27
4.2.3	Hierarchical Prediction through Flat Classification . . . . .	27
4.2.4	Top Down . . . . .	29
4.3	Dataset . . . . .	30
4.4	Evaluation Measure . . . . .	32
<b>5</b>	<b>Experiments Evaluation</b>	<b>34</b>
5.1	Results . . . . .	35
5.1.1	Big Bang with Active Learning . . . . .	35
5.1.2	Single flat classification with Active Learning . . . . .	36
5.1.3	Hierarchical Prediction through Flat Classification with Active Learning . . . . .	36
5.1.4	Top Down with Active learning . . . . .	37
5.1.5	Inter-experiment Comparison . . . . .	38
<b>6</b>	<b>Conclusions and Future Work</b>	<b>40</b>

# List of Figures

2.1	Tree-type hierarchy . . . . .	7
2.2	Directed Acyclic Graph (DAG) type hierarchy . . . . .	8
2.3	Single flat classification approach . . . . .	9
2.4	Hierarchical prediction through flat classification approach . . . . .	10
2.5	Top Down approach . . . . .	11
2.6	Big bang approach . . . . .	11
2.7	Typical Active Learning Setup. Reprinted from [1] . . . . .	14
2.8	A more generalised Active Learning setup. Reprinted from [1] . . . . .	20
4.1	Proposed hierarchy . . . . .	26
4.2	Process flow of ALCC . . . . .	28
4.3	Process flow of ALTD . . . . .	30
5.1	ALBB performance on different seed sets . . . . .	35
5.2	ALSFC performance on different seed sets . . . . .	36
5.3	ALCC performance on different seed sets . . . . .	37
5.4	ALTD performance on different seed sets . . . . .	37
5.5	Cumulative performance on different seed sets . . . . .	39

# List of Tables

4.1	Dataset statistics . . . . .	31
4.2	Dataset class distribution . . . . .	31
5.1	Comparative analysis of the experiments . . . . .	38

# Abbreviations

<b>AL</b>	<b>A</b> ctive <b>L</b> earning
<b>BR</b>	<b>B</b> inary <b>R</b> elevance
<b>DAG</b>	<b>D</b> irect <b>A</b> cyclic <b>G</b> raph
<b>HC</b>	<b>H</b> ierarchical <b>C</b> lassification
<b>MLNP</b>	<b>M</b> andatory <b>L</b> eaf <b>N</b> ode <b>P</b> rediction

# Chapter 1

## Introduction

In theory we mostly work-around with single labeled data through binary or multi-class classification. In binary classification we have choice of choosing among two classes as label for the data instances while in multi-class classification we have to select one label out of more than two possible label outputs. Mitchell's famous machine learning [2] *PlayTennis* problem with *Yes* or *No* as possible classes is an example of binary classification. The same classification example will become multi-class if we add *May Be* as third possible class value.

Real world data often falls in to multiple labels/classes simultaneously, not to be confused with multi-class classification. In multi-class classification still the label is single while in multi-label classification there are multiple labels for the same data instance. Example of multi-label data is a webpage having multiple tags meaning the webpage belongs to all those multiple categories. *Multi-label classification* is the branch of machine learning dealing with the classification of multi-label data. In multi-label classification, an instance can belong to none, some or all of the possible labels in the set of the possible labels. Usually the multiple labels of data instances are not mutually exclusive and bear some intrinsic relation meaning a data instance belonging to a particular category will most probably also belong to some other related categories. For example a document belonging to Fashion will probably also belong to Cosmetics or Boutiques.

At times the multiple labels of multi-labeled data form a hierarchy such that the labels in the upper levels of the hierarchy are more generic as compared to the more specific labels down the hierarchy. Consider the example of a text document belonging to one of the main categories Engineering or Economics, then subsequently having Computer or Civil as sub-category of Engineering or Macro-Economics or Micro-Economics as sub-category of Economics. Web directories like DMOZ <sup>1</sup> is another example of hierarchical categorization where links to websites are arranged in hierarchical categories. *Hierarchical classification*, seen by some as a type of multi-label classification while others see it the other way round as a generalization of multi-label classification, deals with the subject of multiple labels fitting in a hierarchy.

Hierarchical classification has been widely researched in the context of Text classification/categorization, documents classification, images classification, webpages categorization, DNA/Protein function prediction, web directories categorization, medical diagnosis. Wikipedia is a major example where the webpages have been categorized in a hierarchy and the categories are in thousands [3] <sup>2</sup>. Twitter, probably due to short tweet size, has been researched mainly in single label classification perspective.

To develop a reliable model for any classification task, we need a considerable amount of labeled data. Labeling of data has an associated cost in terms of human labelers, time and money. Labeling text documents is more challenging as text documents are usually quite lengthy and takes time to be read and labeled. Tweets are short in length but there are thousands of unlabeled tweets available for every topic and labeling them requires handsome amount of resources. *Active Learning* (AL), a special type of supervised machine learning, is proving useful in the situations where the unlabeled data is abundantly available and we do not have the resources to label all of it. Active Learning makes it possible to develop a good model with only part of the data required to train a supervised learning model, thus requiring less labeling effort and budget.

---

<sup>1</sup><http://www.dmoz.org/>

<sup>2</sup><https://www.kaggle.com/c/lshtc>

To get further insight and applicability of the concept, consider the example of customer reviews related to an electronics item of a company. Customer reviews normally range from thousands to, may be, hundreds of thousands. To analyse these customer reviews we may consider learning a supervised model, for which we have to label a major portion of these customer reviews incurring handful resources in terms of humans, money and time. Instead, Active learning techniques can be utilized to learn an acceptable model with the labeling of nominal number of these reviews thus incurring less labeling cost.

In traditional passive learning, humans supervise the learning process and feed the machine learning algorithm with labeled instances. All of these instances may not be useful for the classification task in hand. Therefore we need enough instances to get a good classification model. In contrast, Active learning empowers the learning algorithm to specify the instances which it consider more contributing than the others for the classification task and let only those be labeled. This makes it possible to achieve acceptable classification accuracy with less (minimal) labeling effort compared to that of supervised learning.

In active learning process, based on the output predictions of the unlabeled data by a meagerly trained machine, some active sampling strategy selects the instances deemed most informative and effective for classification and forward it to human labelers for labeling. Model is retrained and updated with the newly labeled instances to select next batch of most informative and effective instances to be labeled. This process is performed iteratively till an acceptable model is generated ideally having comparable accuracy to that of supervised learning based model.

Hierarchical classification is equally important in the twitter domain as tweets can also be categorized into multiple hierarchical categories. For instance the tweet “@Microsoft Heard you are a software company. Why then is most of your software so bad that it has to be replaced by 3rd party apps?” belongs to the topic Microsoft and subsequently bearing Negative sentiment. On the other hand the tweet “@ScottArbeit @GabeAul @Microsoft isntall the newest version and you may chance your mind!” also belongs to the topic Microsoft but bearing Positive

---

sentiment. Suppose an analyst needs only positive tweets on a particular topic then hierarchical classification can help get the most relevant tweets out of the heaps. To generate models for such hierarchical classification, active learning is a viable solution to minimize labeling cost.

Hierarchical classification of tweets, and active learning as a solution to deal with labeling of abundantly available unlabeled tweets presents an interesting research problem providing motivation for this thesis. There are four major hierarchical classification approaches in literature. We investigated the performance of these four hierarchical classification approaches in Twitter domain complemented by the active learning for reducing labeling task.

The remainder of this thesis is organized such that Chapter 2 provides the background knowledge and theory of hierarchical classification and active learning. Chapter 3 provides design details of our devised experiments. Chapter 4 provides the results and evaluation of the experiments performed. Chapter 5 contains the intended future work and possible enhancements to the present work.



# Chapter 2

## Preliminaries and Background

In this chapter we provide some background knowledge, key concepts and application areas of hierarchical classification and active learning.

### 2.1 Hierarchical Classification

Most of the machine learning research deals with single-labeled binary or multi-class classification. In single-labeled binary case we have to select one of the two possible classes. In case the number of possible classes is greater than two and we are to select one of them then the problem becomes single-labeled multi-class. Most of the classifiers have been devised for single-labeled jobs, even some are inherently suited for binary single-labeled classification like SVM.

Most of the data nowadays is multi-labeled, means it belongs to more than one class under more than one categories. Example of multi-labeled data is a thesis document on the topic of active learning. Thesis on the topic of active learning can be placed in the folder Computer Science, and/or Machine Learning, and/or Data Mining, and/or Hierarchical classification, and/or Active Learning. The multiple labels usually bear some relation and an instance falling under one label may also be belonging to the related categories, as in the above example. The order of the multiple labels may or may not be important. The number of labels an instance

belongs to may range from none to many or almost all the possible labels. The multi-labeling field has gained very much popularity in research communities due to the multi-label nature of the real world data.

Multi-label classification, a branch of machine learning, deals with multi-label classification problems. As per [4], machine learning tackles multi-labeling task in two ways: transforming the multi-label task into multiple single label tasks known as Problem Transformation or transforming (moulding) existing single label classification algorithms to work for multi-labeling tasks known as Algorithm Adaptation. Binary Relevance (BR) is the most common problem transformation method where we consider each one of the multiple labels as a binary classification task and say for “n” labels we train “n” binary classifiers, one for each of the “n” labels. All the “n” classifiers predict the label for an unseen instance and only the labels of the classifiers having a prediction confidence above a threshold are assigned to that instance. As far as algorithm adaptation is considered multi-label decision trees, Rank-SVM, multilabel KNN are few to name modified versions of their original single label versions.

At times the labels in the label set of multi-label data forms a hierarchy such that labels/categories down the hierarchy are more specific as compared to up the order generic categories. Consider the thesis example, a thesis document may generally belong to Computer Science or Mechatronics category. Suppose belonging to category Computer Science the document can further be placed in one of the Machine Learning or Cryptography categories. If belonging to machine Learning the document can further be categorized under Active Learning. In such a hierarchy an instance belonging to a class in the hierarchy will also be belonging to the parent classes of that class in the hierarchy.

Hierarchical classification is special kind of multi-label classification dealing with labels forming a hierarchy. While multi-label classification deals with multiple labels having some relation and preservation of this relation is the major challenge in multi-label classification, hierarchical classification has to take care of the hierarchy and order of the labels as well.

CN Silla jr. et al [5] performed a detailed and comprehensive survey on hierarchical classification and E Costa et al [6] reviewed performance evaluation measures of hierarchical classification. According to both these studies the key considerations in a hierarchical classification task are: type of hierarchy, depth of hierarchy and the classification approach being used. These three considerations are explained in the following sections.

### 2.1.1 Type of Hierarchy

Hierarchies can take the form of a Tree or Directed Acyclic Graph (DAG) . In tree type hierarchies each node has exactly one parent and each node can be reached through one and only one path, as shown in Figure 2.1.

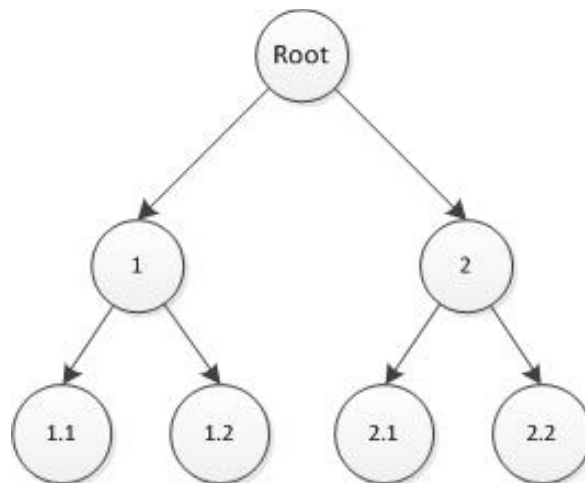


FIGURE 2.1: Tree-type hierarchy

On the other hand in Directed acyclic graph (DAG) a node can have more than one parent meaning a node can be reached through multiple paths in the hierarchy, as shown in Figure 2.2. It is important to mention that as per the qualification of DAG, no circular relation should exist among the nodes.

In hierarchical classification tasks the hierarchy of the labels can be (incrementally) learned from the training data itself at learning phase or it can be hardwired beforehand. For example in [7] the inherent hierarchy of the labels is learned and updated dynamically through concept-drift.

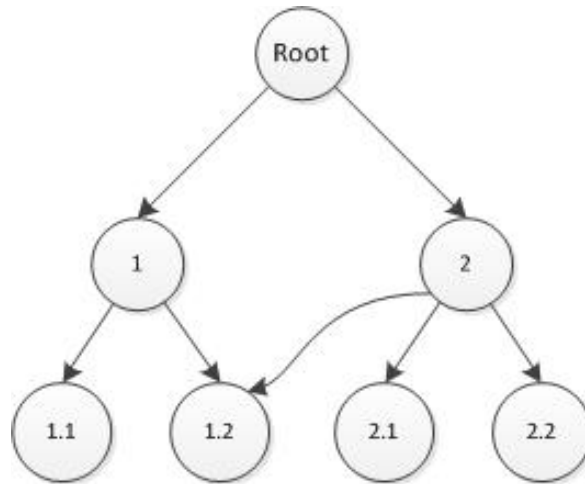


FIGURE 2.2: Directed Acyclic Graph (DAG) type hierarchy

### 2.1.2 Depth of Hierarchy

Some hierarchical classification jobs require the instances to be classified at all the levels till the leaf nodes, termed as “Mandatory Leaf-Node Prediction” (MLNP) problems. For example, a graduate student of Sabanci University must be belonging to one of the faculties and then must be having one of the many faculty members of that particular faculty as thesis advisor. Some tasks may provide the flexibility of discontinuing further classification at any node in the hierarchy and classification till leaf node may not be compulsory, such tasks may be termed as “Optional Leaf-Node Prediction” problems. For example, a document belonging to Computer Science, further belongs to Data Mining but may not necessarily be belonging to sub-categories of Data Mining. In such tasks, the criteria for continuation of further classification down a hierarchy is the qualification of a prediction confidence threshold. If the prediction confidence drops below a threshold value at a node further classification is not useful and discontinued.

### 2.1.3 Classification Approaches

Hierarchical classification has been tackled with four different approaches: single flat classification, hierarchical prediction through flat classification, top-down

classification and big bang (global) classification. A brief overview of these four approaches is provided:

- In *single flat classification* approach only the leaf node classes are taken into consideration, neglecting the rest of the hierarchy, and single label flat classification is performed for these leaf node classes. Single flat classification approach is the simplest among the four. By predicting the leaf classes we automatically predict the ancestor classes. The potential problem with this approach is that we put all our eggs in one basket and misclassification of an instance implies the misclassification of many or almost all of the ancestor classes of that particular instance. For example, in Figure 2.3 an instance belonging to class 1.1 if misclassified by the classifier as class 1.2 implies misclassification of two labels while the same instance if misclassified as class 2.1 implies misclassification of all the labels from leaf node to the top root node.

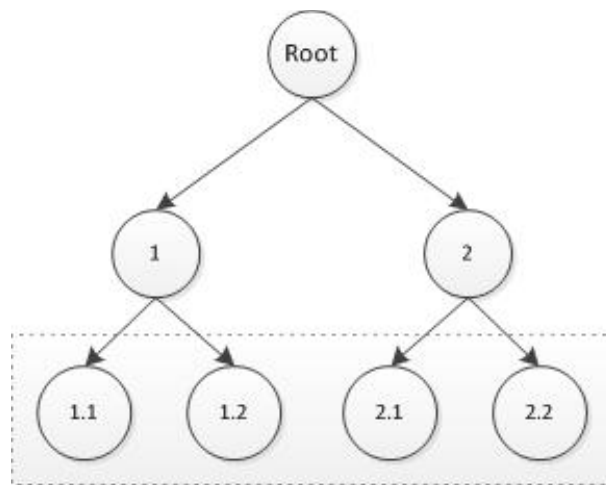


FIGURE 2.3: Single flat classification approach

- In *hierarchical prediction through flat classification* approach we treat each level of the hierarchy as an independent single flat classification problem, neglecting the structure of the hierarchy, and train a single flat classifier for each level, as shown in Figure 2.4. This approach is identical to multi-label classification. The main discrepancy with the approach lies in the fact that the level classifiers are flat therefore labels predicted by the classifiers

may be not consistent with the original hierarchy. For example, an instance belonging to hierarchy  $1 \triangleright 1.1$  can be inconsistently misclassified as  $1 \triangleright 2.1$  or  $2 \triangleright 1.1$ . As far as the complexity is concerned, this approach can be ranked second to the single flat classification approach.

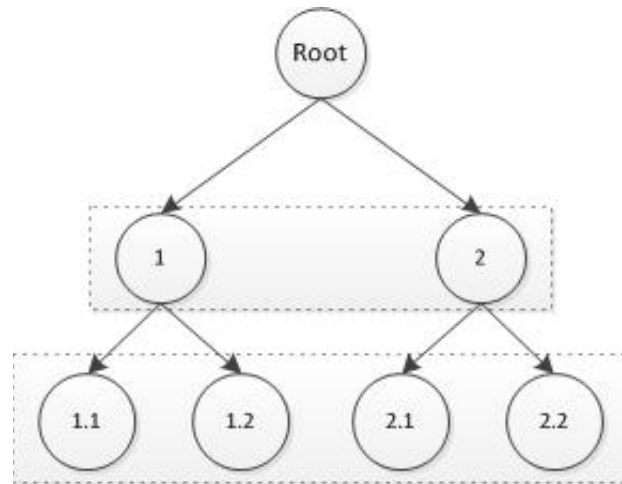


FIGURE 2.4: Hierarchical prediction through flat classification approach

- In *top-down* approach multiple single-label classifiers local to a hierarchy level are trained. The approach can be conceptualized as a tree of classifiers. The root classifier is trained with all the available labeled data. The next level of classifiers are trained with the instances belonging to only one of the parent classes. Referring to Figure 2.5, the root classifier is trained with the whole training data with respect to first label. At level 1 the classifier 1 is trained only with the instances belonging to class 1 of the level 0 and the instances belonging to classes other than class 1 are not included in its training. Similarly classifier 2 is trained only with the instances belonging to class 2 of the level 0 and the instances belonging to classes other than class 2 are not included in its training. Same approach is used for the classifiers at level 2.

In testing phase, the test instance starts predicting its labels from the root node and based on the prediction made is further exposed to the relevant classifier in the level 1 and then level 2 till the leaf node label is predicted. The main discrepancy with this approach is the propagation of the error

made at upper levels down to the lower levels. A test instance misclassified at an upper level will then subsequently be misclassified at all the following levels.

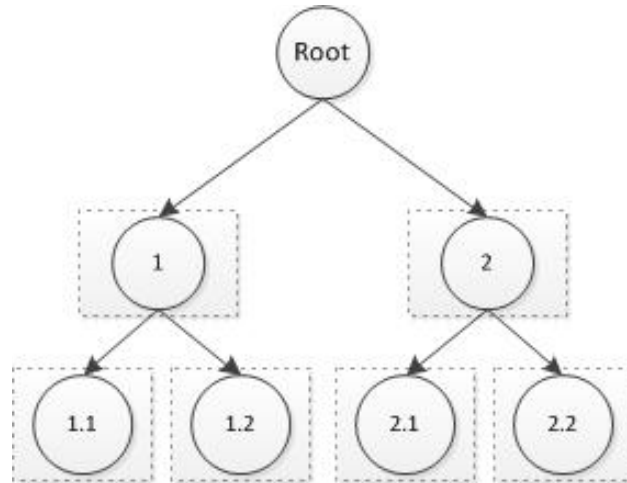


FIGURE 2.5: Top Down approach

- In *big bang* approach we have a global classification model which deals with the hierarchies internally and for user the training and testing is performed in single runs. The implementation complexity lies in the algorithm. The concept is depicted in Figure 2.6

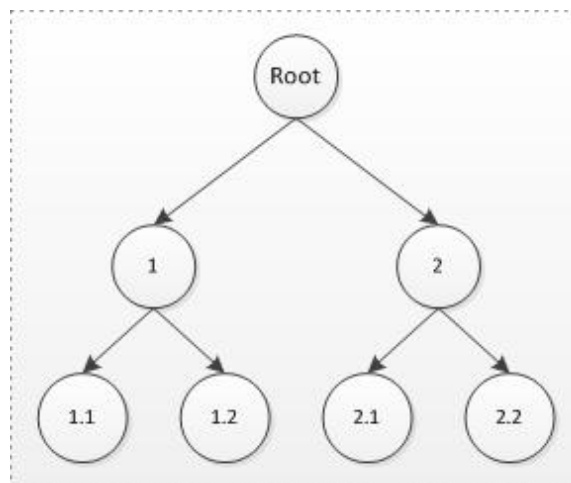


FIGURE 2.6: Big bang approach

### 2.1.4 Applications of Hierarchical Classification

Enormous growth and adoption of electronic documents, necessitated Hierarchical classification in the context of Text categorization so that electronic documents may be categorized on the lines of the conventional files. Along with text documents, digital libraries and emails also need to be hierarchically categorized which is manually impossible. Hierarchical classification is also used in image classification, categorization of ever increasing huge number of webpages, DNA/Genes classification, written text recognition, web directories, diagnosis of fatal diseases and their medication.

Wikipedia is the prime example where hierarchical classification is heavily researched. There are about 325,000 categories and 2,400,000 documents in wikipedia corpus and these categories have a hierarchy<sup>1</sup>. Manual classification is becoming difficult as more categories and documents are being added.

One striking benefit of hierarchical classification is avoidance of ambiguity. For example a thesis document on the topic of active learning will be archived by some in active Learning folder under the umbrella of the machine learning while others may place it in active learning folder under academic Learning. But if hierarchy is followed, the thesis will reach its exact folder.

---

<sup>1</sup><https://www.kaggle.com/c/lshtc>



## 2.2 Active Learning

According to Wikipedia<sup>2</sup>, “Active learning is a model of instruction that focuses the responsibility of learning on learners”. In line with this concept of the academic active learning, the active machine learning puts the responsibility of learning on the learner (learning algorithm). Learner is made incharge of the learning process and the training data being used in the process. The learner selects the data it deems necessary and effective for its learning and the humans (Oracles) label it. Due to the involvement of both human and machine in the process, active learning is also known as “human-in-the-loop machine learning”. Through active learning we can achieve optimum classification performance with comparably less labeled data as that of supervised learning.

To get insight of the concept, consider the example of a Calculus course. Calculus text book contains thousand(s) of questions and it is nearly impossible to solve and teach every question of the text book in the class. Normally the instructors solve a few questions in a chapter as a starter and give rest of the chapter as home assignment to the learners (students). The learners go through the whole chapter and note down the question they deem as difficult and unable to solve. These difficult to understand questions are solved by the instructor in the next lecture. In this way with the coordination of the learner and the teacher, the learner develops a deep understanding of the course with only part of the whole text. On the analogy, in active learning the learner selects the instances it finds hard to predict and ask the teacher (oracle in this case) for labeling the same, resulting in yielding a good model with part of the available data.

In passive learning human is incharge of the learning process and learner behaves like an observer. Human provides the labeled instances to the learner and the learner is trained on these instances. Humans cannot judge the instances necessary for the rightful training of the learner and hence high number of labeled instances is required to learn a good model. In contrast, in active learning the learner itself nominates the instances it deems most important for the classification task

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Active\\_learning](https://en.wikipedia.org/wiki/Active_learning)

for labeling through the humans. Therefore active learning achieves comparable performance to that of supervised learning in fewer instances.

Active learning has been found very useful in the situations where unlabeled data is abundantly available and labeled data is scarce. Labeling has an associated cost in terms of labelers, money and time. So labeling effort is minimized by using active learning.

In a typical active learning scenario, depicted in Figure 2.7, we have a learner (a machine learning algorithm), a labeled set (pool) initially consisting of few labelled instances (ground truth) termed as seed set, an unlabelled pool of instances in a pool based active learning setup or a stream of unlabeled instances in the case of stream-based active learning setup, oracles (humans as domain experts). The active learning process starts with the training of learner on the seed set. The learner then, based on some sampling strategy, selects the most informative instance(s) from the unlabeled pool. The oracle(s) labels the selected instance(s). The labeled instance(s) is/are added to the labeled pool and the learner is updated on the newly labeled instances to select the next most informative instance(s) for labeling, on the basis of updated knowledge. This process continues in an iterative manner till some pre-defined stopping criteria is satisfied.

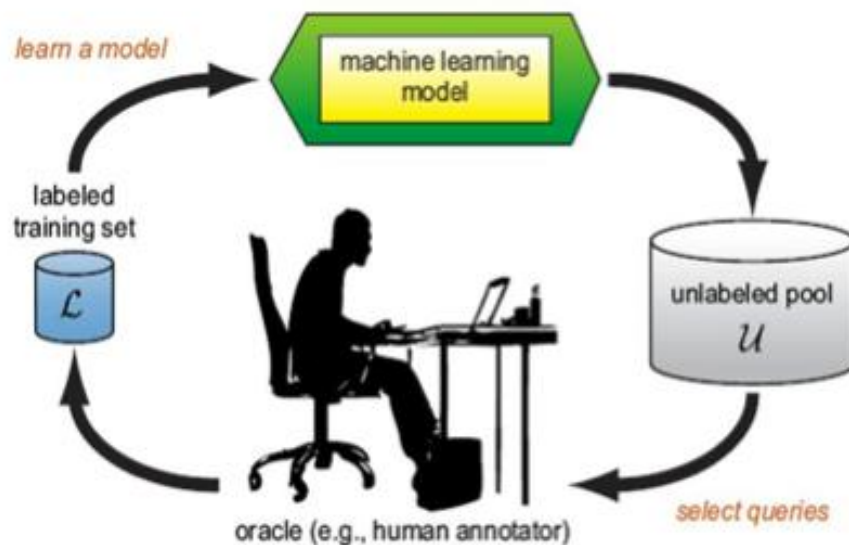


FIGURE 2.7: Typical Active Learning Setup. Reprinted from [1]

In the next subsections we will shed light on the important components of the active learning model and process.

### 2.2.1 Learner

The active learner has a central role in the active learning process. As with supervised learning, the learner in active learning model can be any machine learning algorithm depending on the type and domain of the job. The main job of the active learner is to predict the labels of the unlabeled instances in the unlabeled pool. Based on the difficulty faced or lack of confidence in prediction by the learner, the sampling strategy nominates the most informative instance out of the unlabeled pool to be labeled by the oracle.

Usually, but not always, classifiers capable of yielding prediction probabilities, e.g., Naive Bayes, are used as learners in active learning process for they provide an easy way to measure the informativeness of the unlabeled instances. Support Vector Machines (SVM) are also widely used in active learning experiments as the support vectors and decision boundary of the SVM provides the notion of the informativeness of the unlabeled instances.

### 2.2.2 Initial Seed

Although there are no concrete quantitative figures available about the size or composition of the seed set but as per [8] there are some key guidelines about the selection and composition of the seed set:

- The seed set should be a representative of the classes that the classifier is expected to handle in active learning process. Consider the case where the seed set is missing instances about one of the classes in the distribution. The learner is trained on such seed set and made to predict the labels of the instances in the unlabeled pool. As the learner does not know about one of the classes, the learner will try to fit the instances of that class from the

unlabeled pool to the classes it knows through seed set and hence that class may be left out in the active learning process and the model will not be able to predict that class.

- Ideally the class distribution in the seed set should be kept the same as that of the unlabeled data so that model is trained in the right proportion of the classes.
- The size of the seed set has a direct relation with the number of classes the model is going to deal. The greater the number of the classes in the data, the greater number of instances is required in the seed set to train a reliable model.

The formation of seed set is a subjective matter and different approaches have been researched. Some works like [9] suggests to perform clustering prior to active learning and select the instances at the centre of clusters as members for seed set. Unavailability of labeled instances for seed set is known as *cold start problem*. As a simple approach the active learning experiments may be performed multiple times with random selection of seed sets and finally averaging the results.

### 2.2.3 Sampling Strategies

Perhaps the most important and critical component of the active learning process is the instances sampling strategy. The effectiveness of the active learning depends on the instances being sampled which are labeled by the oracle(s) and used to retrain model to select the next batch of to-be labeled instance(s). Effective sampling can lead to a better model with few instances. The goal is to select the most informative instances with respect to the job in hand. The informativeness of the instances is measured through some utility metric. We are discussing few strategies below:

- *Random Sampling*: The instances to be labelled are selected pure randomly without any utility metric calculation. So far overall random sampling has

been found to be the most consistent performer across all domains. Random sampling is considered as benchmark and for any active learning sampling strategy to be accepted and adopted must atleast outperform random sampling.

- *Uncertainty Sampling*: Perhaps mostly used, uncertainty sampling uses the uncertainty of the classifier in predicting the label of an instance as utility metric. Of the unlabeled pool, the instances about whose class/label the classifier is most uncertain are considered the most informative instances and nominated for labeling task. The uncertainty is usually calculated through confidence, margin and entropy using the posterior probabilities of the classifier.
  - Confidence is measured as the highest posterior of the classifier for all the possible classes of a given instance. Instance(s) having the smallest confidence are considered the most uncertain and ideal for labeling.

$$x_{LC}^* = \underset{x}{\operatorname{argmax}} \quad 1 - P_{\theta}(\hat{y}|x)$$

- Margin is the difference of the two highest posteriors for a given instance. Instance(s) having the smallest margin is/are the most uncertain and ideal for labeling.

$$x_M^* = \underset{x}{\operatorname{argmin}} \quad P_{\theta}(\hat{y}_1|x) - P_{\theta}(\hat{y}_2|x)$$

- Entropy provides the notion of incompleteness in various fields. In active learning, entropy takes all the posteriors for an instance into account and through Shannon-Entropy calculates the uncertainty of the instances.

$$x_H^* = \underset{x}{\operatorname{argmax}} \quad - \sum_{i=1}^Y P_{\theta}(\hat{y}_i|x) \log P_{\theta}(\hat{y}_i|x)$$

- *Semi-Random Sampling*: In semi-random sampling, switching between random and uncertainty sampling is iteratively performed. As an example scenario, one instance is randomly selected to be labeled and model retrained to select the most uncertain instance to be labeled in an iterative manner. Through semi-random sampling we can reap the partial benefits of both random and uncertainty sampling. Consider an example scenario where as per the uncertainty based sampling most of the uncertain examples belongs to the same class then the trained model may be biased. Semi-random can avoid the situation by providing chance to the neglected class instances to be labeled and become part of model training.
- *Data Exploration Based Sampling*: Apart from the individual label of an instance, the distribution of data or neighbourhood instances can also be taken into consideration while selecting an instance for labeling. For example, [10] performs clustering prior to active learning and representative instances of clusters are selected for labeling. [11] and [12] devised strategies which takes into consideration the labels of the labeled instances in the neighbourhood of a candidate instance for calculation of its effectiveness. Relying purely on the distribution of data is known as Data Exploration and relying completely on the learner to sample instances for labeling is known as Model Exploitation. Purely relying on exploration or exploitation is not suggested as each has its own drawbacks.
- *Query by Committee*: Instead of single, multiple different learners are trained and used to predict the labels of the unlabeled instances in the unlabeled pool. Disagreement by the classifiers over the labels is used as the utility metric and the instances bearing maximum disagreement are deemed to be most uncertain and nominated for labeling.

### 2.2.3.1 Synthetic Queries

In some cases, for training a model synthetic queries may be generated by the learner and labeled by the oracles, as shown in Figure 2.8. The synthetic queries

are generated by altering attributes of the queries known to the learner. The concept may be useful as the model is trained on breadth of instances but almost useless in some cases like in case of text categorization. Just altering the attributes may result in the generation of grammatically wrong or incomplete queries and may completely make no sense to the oracle.

### 2.2.3.2 Batch-mode Queries

Our active sampling strategy may select a single most informative instance to be labeled or it may select a batch of multiple queries. Batch mode selection of queries speeds up the active learning process as the assessment of informativeness of instances takes time but on the other hand some of the instances in the batch may not be required to be labeled if some other queries in the batch are labeled, hence wastage of labeling budget.

## 2.2.4 Unlabeled Data (Pool)

In pool-based active learning scenario, we have a large pool of unlabeled instances from which the sampling strategy, through the learner, iteratively selects the most informative instances to be labeled by the oracle. While in stream-based setting, one instance a time arriving in a stream is dealt and evaluated to be enough informative to be labeled or discarded. The scenarios are depicted in Figure 2.8.

## 2.2.5 Oracles

It is usually assumed that oracles are perfect humans, experts in the problem domain or at least domain-aware, reliable and consistent performers. Labeling can be done in single-instance querying with single-oracle environment or multi-instance querying with multi-labelers environment. In case of hierarchical classification, oracle(s) per hierarchy level may be required as a single oracle may not be expert of every category. Recently crowd-sourced labeling [13] has gained popularity

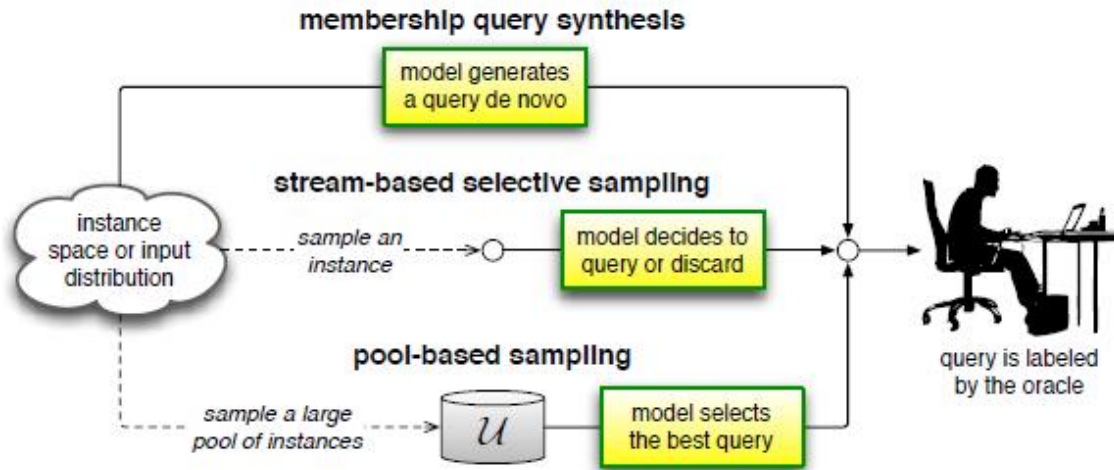


FIGURE 2.8: A more generalised Active Learning setup. Reprinted from [1]

and some platforms like Mechanical Turk<sup>3</sup> and CrowdFlower<sup>4</sup> provide services for crowd-sourced labeling. As the labels acquired from such a setting may not be completely reliable therefore strategies have been suggested for coping with such noisy labels. For example, if the labeling accuracy of a labeler falls below an acceptable threshold the labeler may be refrained from further labeling. [14] suggest strategies where the queries considered hard to be labeled by the labelers are labeled by the domain experts.

## 2.2.6 Stoppage Criteria

“When are we supposed to stop active learning?”. This is fairly an objective question. As per [8], various approaches have been devised for catering this question. The simplest approaches may be: whenever (a predefined) target accuracy is achieved or a fixed number of iterations are completed or the unlabeled pool is exhausted then stop further learning. One more intuitive approach is to monitor the performance measure and whenever a decline in performance is noticed stop the learning. Another simple approach is that if cost of labeling further instances exceeds the misclassification cost then active learning can be stopped.

<sup>3</sup><https://www.mturk.com/mturk/welcome>

<sup>4</sup><http://www.crowdfunder.com/>



## 2.2.7 Applications of Active Learning

Being proven effective and beneficial for the machine learning, active learning has been used in a variety of applications and domains. Text classification/categorization is perhaps the top beneficiary as labeling long text documents are hard to read and label. Webpages categorization, images classification are favorable for active learning as unlabeled instances in these areas are widely available. Some interesting applications of active learning include Automatic Classification of Software Behavior [15], recommender systems [16], Automatic speech recognition [17], Natural language parsing [18], music retrieval [19], anomaly and rare-category detection [20].

# Chapter 3

## Research Problem

In this chapter we will provide the details of the research problem we have investigated. An overview of the related work performed is provided.

### 3.1 Problem Definition and Proposed Approach

Twitter has not been thoroughly researched in the context of hierarchical classification. Hierarchical classification of tweets is equally important as other domains. For example an electronics company may want to categorize the pool of tweets about their different products into positive and negative user feedback. With hierarchical classification, they will be able to categorize and then analyse the users feedback about different products. Existing hierarchical classification approaches devised mainly for text classification can be validated in Twitter domain. For reducing the labeling effort required for training these models, active learning can be utilized.

We tailored the four existing hierarchical classification approaches with active learning to be used for hierarchical classification of tweets. The big bang approach will be used as benchmark for comparison of the other experiments.

## 3.2 Research Hypotheses

Through the devised experiments we will try to satisfy the following hypotheses:

- Conventional hierarchical classification approaches can be utilized for hierarchical classification of tweets.
- Sophisticated hierarchical prediction through flat classification and top-down approaches are expected to perform better than the rest of the two approaches.
- As proved in other domains, active learning can be used for hierarchical classification in twitter domain, thus saving considerable amount of labeling effort.

## 3.3 Related Work

As discussed in previous sections, Twitter has not been able to gain researchers attention in the context of Hierarchical Classification. To the very best of our knowledge, only Zhaochun Ren et al [7] researched hierarchical classification in Twitter domain. To make up for the short length of the tweets, they are extending the short text of tweets using wikipedia corpus. They used simple supervised learning for training the model.

Regarding the application of active learning in hierarchical classification, we know about two works [21] and [22]. Both these works are in text classification domain and not in twitter domain. Both these works use top-down approach for hierarchical classification. Performance comparison of their approach is made with random sampling.

[21] is using global unlabeled pool and each node is performing individual active learning in parallel fashion for itself irrespective of the other nodes. The problem

with the approach is the selection of irrelevant instances. If an irrelevant to the node instance is selected to be labeled the budget is wasted.

Xiao Li et al [22] enhanced Xiao Li et al [21] by introducing individual unlabeled pools for each node consisting of only the instances relevant to one of the parent class. Active learning is performed for root node and then refined relevant unlabeled pools are constructed for subsequent nodes from the perfectly labeled instances by oracle. As the size of the unlabeled pools keeps shrinking down the hierarchy, the authors proposed a novel approach of using trained classifiers in the upper levels to label the unlabeled instances in their unlabeled pools and these instances are used to populate the unlabeled pools of the subsequent nodes. To cope with the misclassifications performed in these labeling and reduce their effect, dual-pool strategy has been suggested. One of the unlabeled pool contains the instances perfectly labeled by oracles and the other unlabeled pool contains the (somehow noisy) instances labeled by the classifiers which may contain instances misclassified by the upper level classifier and being irrelevant to the concerned node. Based on some heuristics, instances are selected by the active sampling strategy in some proportion out of these pools.

Sufficient research has been performed in hierarchical classification of text therefore benchmark performance for comparison exists. For Twitter domain as there is no work so we are doing the comparative analysis of all four approaches. We are using a simplified version of [21] and [22], by deploying a single unlabeled pool instead of the dual-pool strategy.

# Chapter 4

## Experiments Design

In this chapter we explain the experiments designed for comparing the four hierarchical classification approaches. Details of the hierarchy considered for the experiments, dataset and evaluation measure used for comparison of the experiments are provided.

### 4.1 The Hierarchy

For our hierarchical classification experiments, we are using the pre-defined hierarchy provided as Figure 4.1. We have three levels in the hierarchy. At root level we have two classes: Sports&Entertainment and Politics. At second level of the hierarchy we have two classes: Factual and non-factual for each of the two parent classes. At third level in the hierarchy we have Positive and Negative classes in case of non-factual parent class or Neutral in case of factual case. Looking at the hierarchy, classification till the leaf nodes is required and hence the task is Mandatory Leaf-Node Prediction (MLNP) problem.

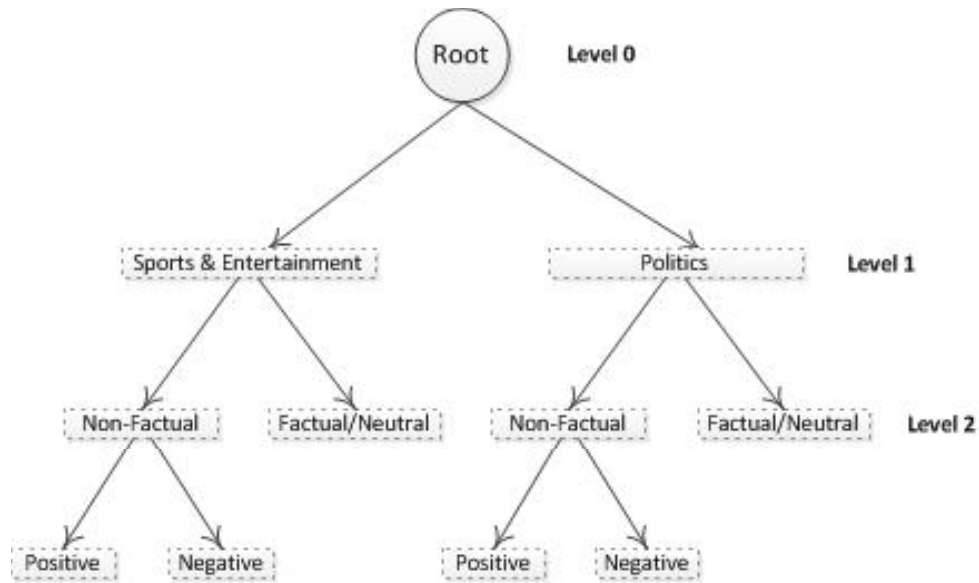


FIGURE 4.1: Proposed hierarchy

## 4.2 Experiments

### 4.2.1 Big Bang

Our first experiment is based on the concept of big bang approach of hierarchical classification. The experiment is carried out through MEKA. MEKA framework is WEKA's extension for multi-label classification<sup>1</sup>. MEKA contains both the problem transformation and algorithms transformation methods for multi-label classification and multi-target classification. The interface and operation of MEKA is very much identical to that of WEKA. As the spirit of the big bang approach, MEKA looks after the implementation of the multi-labeling tasks and for users the job is performed in a single run. It is important to mention here that MEKA performs flat classification means lacking hierarchical classification.

From now onwards we will call this experiment ALBB. In ALBB, we use the Class-Relevance (CR) class of MEKA multi-target classification. CR is the generalised multi-target version of the Binary Relevance (BR) method of multi-label classification, meaning the relation among labels will be neglected and multiple classifiers will be trained for multiple labels.

<sup>1</sup><http://meqa.sourceforge.net/>

For active Learning in ALBB we are using Random sampling for instance selection i.e., instances are pure randomly selected from the unlabeled pool and after being labeled added to the labeled pool until the budget is exhausted. Eventually the model is trained on the labeled pool to perform the classification of tweets in the testset. We will use ALBB as performance benchmark for the other three experiments.

### 4.2.2 Single Flat Classification

As discussed in the previous chapter, we consider only the six leaf node classes and a single flat classifier is trained. Nodes in the hierarchy other than the leaf nodes are neglected as each class represents a full path in the hierarchy. We will call this experiment ALSFC in the rest of the text.

The hierarchical task is reduced to a simple one level multi-class classification task. There is a single unlabeled pool and we select most uncertain instances from this unlabeled pool and ask oracle for labeling of these instances. Labeled instances are added to the labeled pool and model retrained for further selection of uncertain instances till the budget is exhausted.

### 4.2.3 Hierarchical Prediction through Flat Classification

Devised by Jesse Read et al [23] for multi-labeling tasks, classifier chains is a variant of Binary Relevance (BR) in which association among the labels is preserved by adding the preceding label(s) as attribute(s) to the dataset for learning the next label using flat multi-label classification. Being a problem transformation technique, the dataset is transformed into “n” single label datasets, where “n” is the number of labels and “n” numbers of classifiers are trained. Each classifier is trained on one of the “n” datasets such that dataset at position “j” uses the labels of the “j-1” datasets as additional attributes bearing a binary value of “1” in case of instance being positive for corresponding label or “0” in case of negative instance.

As our third hierarchical classification experiment, we used classifier chains concept with local classifier per level experiment. From active learning point of view, we are dividing the budget equally in the three levels. We will be using the term ALCC as identifier for this experiment in the remaining text. In ALCC, we used three local classifiers for each level of the hierarchy. The label output of level 0 is added as an additional attribute in level 1 dataset and both level 0 and level 1 labels output as additional attributes in level 2 dataset.

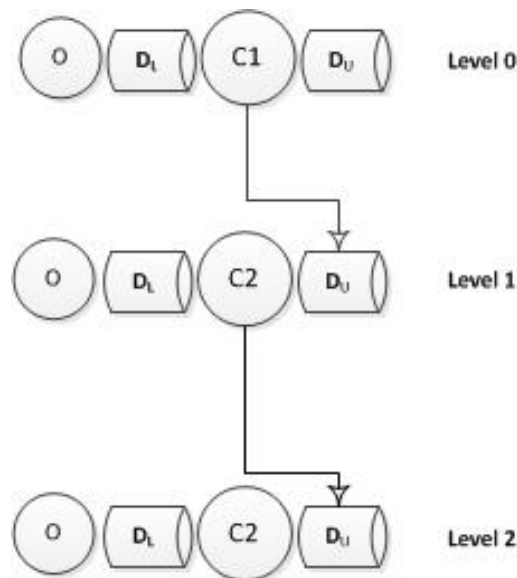


FIGURE 4.2: Process flow of ALCC

For active learning we need a labeled pool initially consisting of the seed set and an unlabeled pool. Referring Figure 4.2, in ALCC we are having three classifiers “C” one for each level and every classifier having its unlabeled pool “ $D_U$ ”. As we are using Classifier Chains concept, the unlabeled pools for level 1 and level 2 are empty initially. Active learning is performed for the first level and most uncertain instances from the unlabeled pool are labeled by oracle “O” till the exhaustion of share in the budget. These labeled instances are added to the level 0 labeled pool “ $D_L$ ” as well as added to the level 1 unlabelled pool having level 0 labels appended as attribute. Once the budget share is consumed, the level 0 trained classifier is then used to predict the labels of the instances left out in the unlabeled pool of the level 0 to make sure provision of sufficient unlabeled instances in the unlabeled pool of the level 1. For avoiding noise, we only considered the labels for which



the classifier was atleast 80% confident to be placed in the level 1 unlabeled pool. Similarly, active learning is performed for the level 1 and the labeled instances are concomitantly added to the unlabeled pool of the level 2 having both level 0 and level 1 labels appended as attributes. After the exhaustion of the level 1 budget, the learned classifier is used to label the left out instances in the unlabeled pool of the level 1 and instances qualifying 80% confidence criteria are placed in level 2 unlabeled pool. Finally, active learning is performed for the level 2 and classifier trained.

In testing phase, the instances are exposed to the three level classifiers in turn and labels predicted.

#### 4.2.4 Top Down

As our fourth experiment, we used top down approach. We will be using ALTD as identifying term for this experiment. We used a local classifier for each node approach. Each classifier is trained with only the instances belonging to one of the parent class. It is important to mention here that budget is equally distributed among the levels and the level budget is equally distributed in the classifiers in a level.

Referring Figure 4.3, in ALTD we are using individual unlabeled pool " $D_U$ " for each classification node. The unlabeled pool of each classifier should only contain the instances belonging to one of the parent class. On the lines of the ALCC, active learning is performed at level 0. The labeled instances are added to the level 0 labeled pool " $D_L$ " as well as to the unlabeled pool of node 1 or node 2 at level 1 depending on the label provided by oracle " $O$ ". After the due budget share for level 0 is consumed, the learned classifier is used to predict the labels for the left out instances in the unlabeled pool of the level 0 and sent to respective unlabeled pools of the level 1 depending on the predicted labels. For avoiding noise, we only considered the labels having atleast 80% classifier confidence to be placed in the next level unlabeled pools. Similarly, active learning is performed for both the

nodes at level 1 and the labeled instances are sent to the respective labeled pools as well as the unlabeled pools in the level 2 depending on the labels provided by the oracle. After the exhaustion of the due share in budget, the left out instances in each unlabeled pool at level1 are labeled by their respective learned classifiers and the based on the predicted labels the instances are sent to the unlabeled pools at level 2. Eventually active learning is performed at level 2 and classifiers trained.

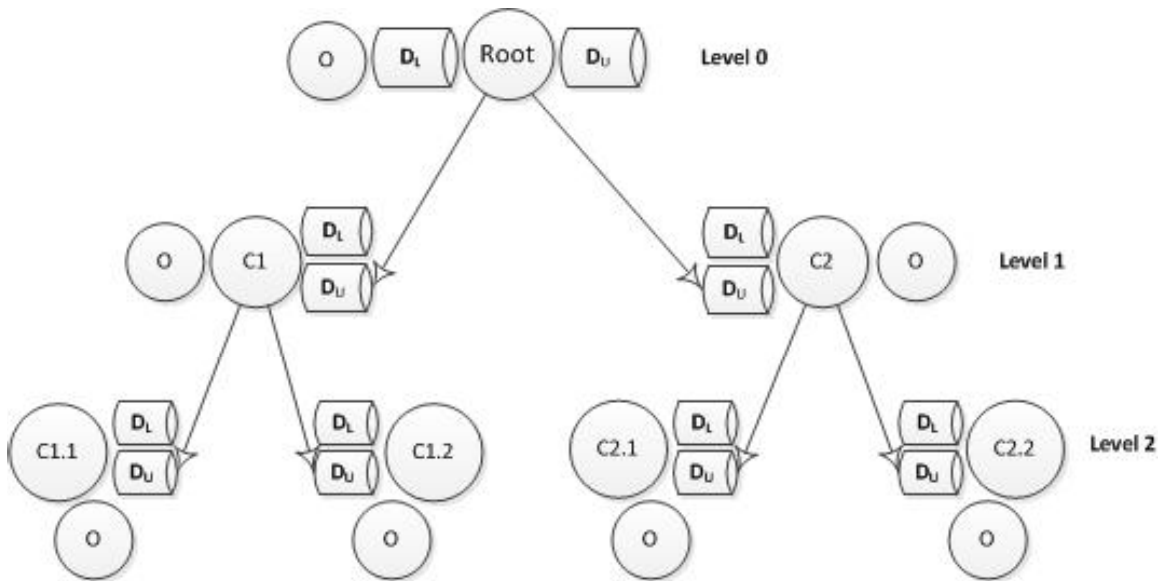


FIGURE 4.3: Process flow of ALTD

Testing is also performed in the top down fashion. A test instance is classified by the root node and based on the predicted level 0 label the instance is further exposed to the relevant node and this goes on till the leaf node is predicted.

### 4.3 Dataset

For conducting the experiments, we used SemEval twitter dataset [24],[25]. SemEval is an international workshop on semantic evaluation<sup>2,3</sup>. The original dataset is a collection of tweets on eighty (80) different topics. We selected tweets related to forty three (43) topics and manually transformed them to three-level proposed hierarchy. Originally the tweets were bearing two labels: one for topic and other

<sup>2</sup><http://alt.qcri.org/semeval2016/1/>

<sup>3</sup><https://en.wikipedia.org/wiki/SemEval>

for the sentiment of the tweet i.e., Positive, Negative or Neutral. We treated the topics as first label and merged them to two broad categories. The sentiment was transformed to two labels. Positive and Negative were considered as non-factual second label and Neutral as factual. The sentiment itself is considered as third label. Composition of the final dataset is provided in Table 4.1.

Dataset	Instances	Features	Hierarchy Level
SemEval	4332	10500	3

TABLE 4.1: Dataset statistics

The class distribution of the final dataset is provided in Table 4.2. The dataset is skewed.

Level 1 Categories	Level 2 Categories	Level 3 Categories
Sports&Entertainment (3041)	Non-factual (2161)	Positive (1948)
		Negative (213)
	Factual/Neutral (880)	
Politics (1291)	Non-factual (716)	Positive (283)
		Negative (433)
	Factual/Neutral (575)	

TABLE 4.2: Dataset class distribution

As we know, due to 140 characters limit and informality of the medium, tweets contain informal language, extensive use of abbreviations, typos and special characters. To be fit for classification tasks, tweets need to be preprocessed. Following steps were taken to clean and preprocess the tweets data:

- *Removal of URLs:* Tweets heavily contains URLs to other pages. These links do not contribute to the classification task hence they needs to be removed.
- *Removal of Punctuation:* Extensive punctuation is used by users. These punctuations make complex the language hence are removed.

- *Removal of Numbers*: Numbers and dates are less important for classification and therefore were removed.
- *Removal of StopWords*: Stopwords do not contribute to the classification tasks and uselessly increase the number of attributes therefore we removed stopwords using NLTK stopwords list.
- *Stemming*: Where possible we stemmed the words, like *work*, *works*, *worked*, *working* were stemmed to *work*.

We use simple Bag-of-Words (BOW) representation for tweets in our classification task. Due to the reason mentioned in the next section, the preprocessed tweets coupled with their respective labels were saved in ARFF format as both WEKA and MEKA uses ARFF file format.

Using Bag of Words approach for attributes in text classification makes it different from other classification jobs. In other classification jobs the attributes of the training set and test set are always the same, only the values of those attributes changes. In text classification the number of attributes and the attributes itself may differ in training set and testset. To handle the situation, we chain the indexing process and the classifier. We used WEKA's Filtered Classifier to chain String to Word Vector filter with the classifier to perform the indexing of the data on-the-fly.

## 4.4 Evaluation Measure

Some researchers use flat classification evaluation measures for performance assessment of hierarchical classification while many devised their own measures suitable to their machine learning domain. A comprehensive comparison of different performance measures has been carried out by [6] and [26]. Ideally, a hierarchical classification task should be evaluated with hierarchical evaluation measures.

Being most advocated, we are using hierarchical F-measure for evaluation of our hierarchical classification task. Hierarchical F-measure is a tweaked version of F-measure for hierarchical evaluation calculated using tweaked versions of precision and recall. The *hierarchical F-measure* is calculated through Equation 4.1.

$$hF = \frac{2 * hP * hR}{hP + hR} \quad (4.1)$$

where

$$hP = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{P}_i|}$$

and

$$hR = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{T}_i|}$$

In the above equations  $hP$  represents hierarchical precision and  $hR$  represents hierarchical recall.  $\hat{P}_i$  is the set of labels predicted for a test instance  $i$  for all the levels in the hierarchy and  $\hat{T}_i$  is the set of true labels for all the levels in the hierarchy of that test instance  $i$ .

Calculation of hierarchical F-measure is simple for ALBB, ALCC and ALTD. In ALSFC approach, we make the hierarchical classification problem a single flat classification problem but to make a comparison with the other three we have to calculate the hierarchical F-measure. As each label represents a path in hierarchy, so each predicted label should contribute the way it contributes in a hierarchical problem. For example, a Positive leaf node class instance of category Sports&Entertainment if misclassified as Negative should be less harmful than the same instance if misclassified as Neutral.

# Chapter 5

## Experiments Evaluation

The designed experiments were conducted with the same parameters and assumptions as follows:

- Margin-based uncertainty sampling, due to its simplicity and wide usage in active learning experiments, is used as utility metric to select most informative instances in case of ALSFC, ALCC, and ALTD. For ALBB we used random sampling.
- Naive Bayes, due to its scalability, inherently providing probability distribution on the instances, proven better results with text classification, is used as the base classifier in all the experiments.
- Each label is assumed to consume one unit of budget. So if all the three labels were provided by oracle then three units of budget is consumed.

To evaluate the models, 10% of the total 4332 instances i.e., 433, were used as Testset while the remaining 90% instances i.e., 3899, were considered as training set. The training set was distributed among seed set and unlabeled pool. Experiments were conducted for three different values of seed sets i.e., 0.1% of training set, 0.5% of training set and 1% of the training set. To exclude randomness effect, each experiment is performed 10 times with prior randomization of the dataset

using unique random seed values. We used 500 budget limit with recording of performance on intervals of 25 instances.

## 5.1 Results

We first present individual results of experiments initiated with the three different seed sets and then inter-experiment comparative results are provided.

### 5.1.1 Big Bang with Active Learning

Figure 5.1 provide the hierarchical F-measure curves of the Big Bang approach with active learning, referred to as ALBB, on three different values of seed sets. As evident, greater the size of the seed set greater the hierarchical F-measure we get. And greater the performance of the seed set lesser the gap for improvement. We can see steepest curve in the case of 0.1% seed set while at 1% seed set the curve is not so steep.

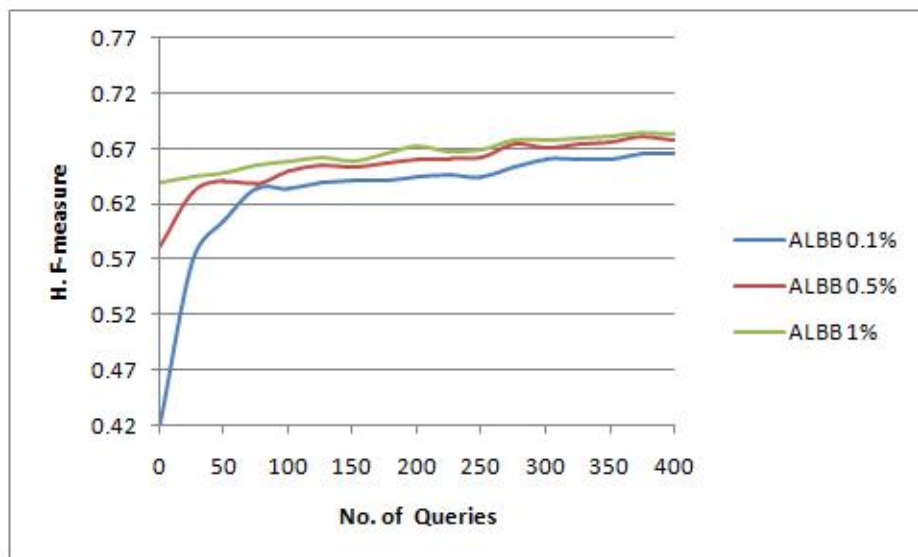


FIGURE 5.1: ALBB performance on different seed sets

### 5.1.2 Single flat classification with Active Learning

In case of Single flat classification with Active Learning, referred to as ALSFC, as expected we get greater performance on larger size of seed set. In Figure 5.2 the three curves converge at budget 150 and remains steady till the exhaustion of budget.

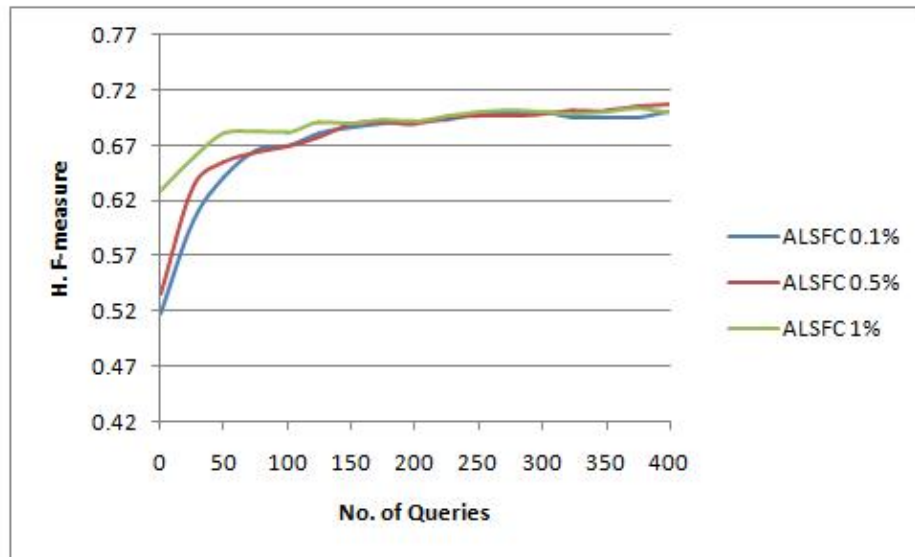


FIGURE 5.2: ALSFC performance on different seed sets

### 5.1.3 Hierarchical Prediction through Flat Classification with Active Learning

Figure 5.3 shows that the hierarchical F-measure performance with 0.1% of seed set is nominal as compared to the other two seed sets hence leaving scope for greater learning therefore we see a continuous rise in performance. The other two seed sets achieve good performance at seed sets and the scope for learning is less as compared to the 0.1% seed set. The performance enters a steady state at the exhaustion of budget.



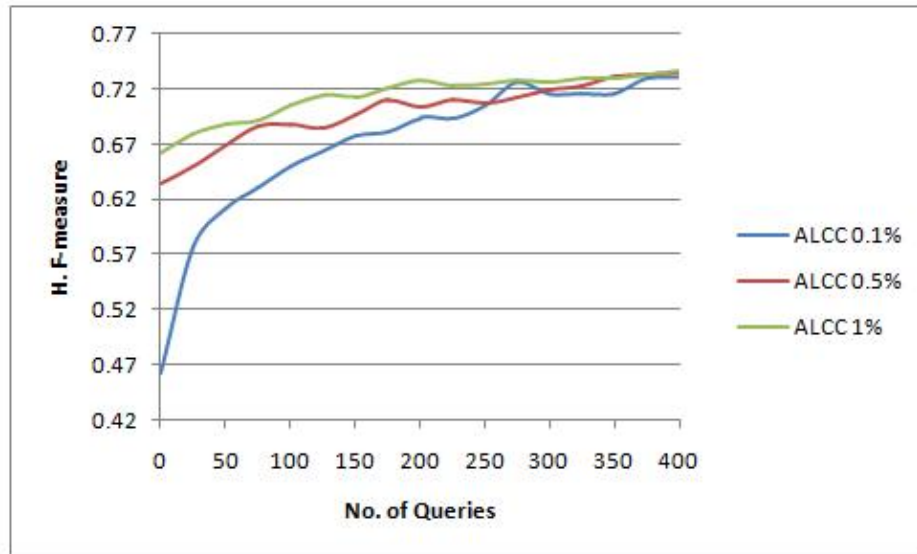


FIGURE 5.3: ALCC performance on different seed sets

#### 5.1.4 Top Down with Active learning

As depicted in Figure 5.4, our last experiment bears significant difference in the performance on the seed sets and the model continuously improve its learning till a steady state is reached at budget 250.

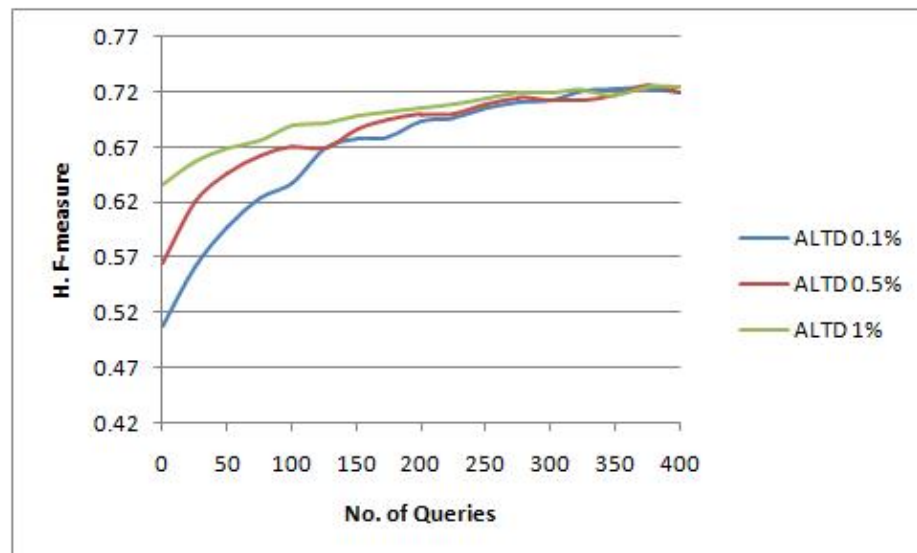


FIGURE 5.4: ALTD performance on different seed sets

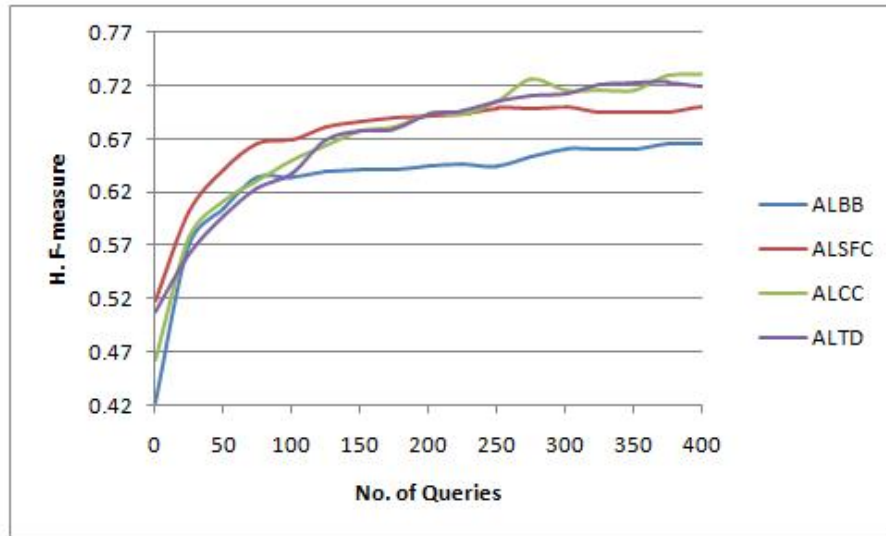
### 5.1.5 Inter-experiment Comparison

Referring Figures 5.5(a), 5.5(b) and 5.5(c), we can see that ALCC (hierarchical prediction through flat classification) is outperforming the other three approaches. In our opinion, ALTD is a good approach but as the data is skewed and budget is equally distributed among the nodes, the nodes with the major data did not get enough share of budget to train enough uncertain instances. As the case with ALCC, budget is used by the level classifiers according to the uncertain instances so the greater the number of instances of a class greater the budget used for that class and hence higher the hierarchical F-measure.

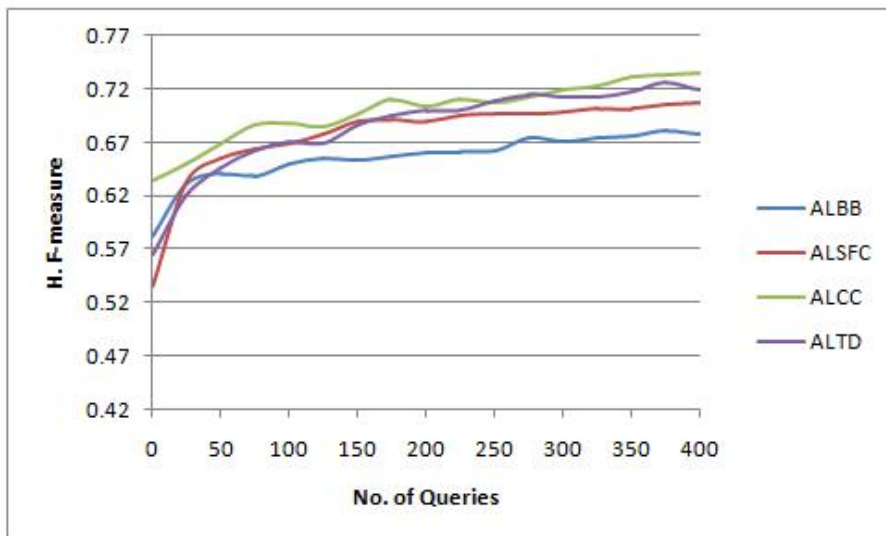
In active learning experiments the effectiveness can be quantified by the number of queries an experiment saves in achieving the best performance of the random sampling counterpart. In our case random sampling based ALBB is the benchmark. Referring Table 5.1, ALBB achieves 0.67, 0.68 and 0.69 hierarchical F-measures on consumption of 400 (maximum allowed) budget units with 0.1%, 0.5% and 1% seed sets respectively. ALSFC outperforms the other competitors in 0.1% case by achieving 0.67 hierarchical F-measure with consumption of 85% less queries as compared to ALBB, but we can see in Figure 5.5(a) that after training with consumption of 250 queries ALCC takes the lead. In 0.5% and 1% seed sets cases, ALCC is leading the competition by saving 85% and 90% queries respectively to achieve the 0.68 and 0.69 hierarchical F-measure of ALBB.

Seed Set	Experiment	H. F-measure	Queries Consumed	Queries Saved
0.1%	ALBB	0.67	400	--
	ALSFC	--	75	<b>85%</b>
	ALCC	--	130	74%
	ALTD	--	125	75%
0.5%	ALBB	0.68	400	--
	ALSFC	--	140	72%
	ALCC	--	75	<b>85%</b>
	ALTD	--	145	71%
1%	ALBB	0.69	400	--
	ALSFC	--	110	78%
	ALCC	--	50	<b>90%</b>
	ALTD	--	90	82%

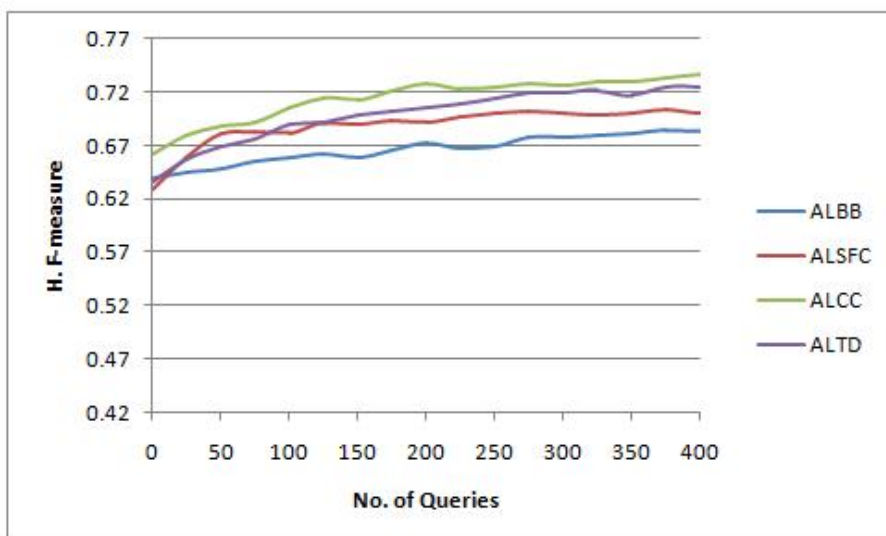
TABLE 5.1: Comparative analysis of the experiments



(a) 0.1% seed set



(b) 0.5% seed set



(c) 1% seed set

FIGURE 5.5: Cumulative performance on different seed sets

# Chapter 6

## Conclusions and Future Work

In this thesis, we used active learning for developing models for hierarchical classification of tweets. For the purpose we devised four models having different underlying concept. The first model is based on big bang approach of hierarchical classification. Second model is based on single flat classification of hierarchical classification. Third approach is based on local classifier per level approach and the third model is based on local classifier per node approach. To reduce labeling task, which requires resources like humans, time and money, we used active learning to select a modest number of instances out of the available pool to be labeled and acquire optimum learning. We used the Hierarchical F-measure as evaluation metric for the experiments was used.

From the results of the experiments, hierarchical prediction through flat classification approach outperformed the counterparts, while all outperforming the random sampling based big bang approach.

Currently labeling budget is equally distributed and consumed by all the levels, and all the classifiers within a level. We will work on deploying schemes for utilization of variable budget by levels and classifiers to achieve better performance or atleast get the classifiers lagging behind at far with the others.

As we are using the learned classifiers to predict the remaining unlabeled instances for next level unlabeled pools, noise is added in case of misclassified instances.

---

Enhancement to the present work is possible in the form of the usage of the oracle feedback for back-propagation of noisy instances by correcting their previous labels and sending them to their respective unlabeled pools.

In present implementation we are using simple bag of words approach for attributes. Additional sophisticated attributes can be added to the dataset to increase classifiers learning and performance as a result.

Query by Committee (QBC) can be used to sample most informative instances and predict labels for the left out instances in unlabeled pools for next level unlabeled pools.

Presently we use uniform budget consumption for each level label, while in real life different labels can incur different cost like based on labeling difficulty or the labeling choices available hence variable cost for different labels can be researched.

# Bibliography

- [1] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.
- [2] Tom M. Mitchell. *Machine learning*. McGraw Hill series in computer science. McGraw-Hill, 1997. ISBN 978-0-07-042807-2.
- [3] Jean Feng, Chuan Yu Foo, and Yifan Mai. Automated categorization of wikipedia pages.
- [4] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 26(8): 1819–1837, 2014.
- [5] Carlos N Silla Jr and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.
- [6] E Costa, A Lorena, ACPLF Carvalho, and A Freitas. A review of performance evaluation measures for hierarchical classifiers. In *Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop*, pages 1–6, 2007.
- [7] Zhaochun Ren, Maria-Hendrike Peetz, Shangsong Liang, Willemijn van Dolen, and Maarten de Rijke. Hierarchical multi-label classification of social text streams. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 213–222. ACM, 2014.

- 
- [8] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.
- [9] Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Off to a good start: Using clustering to select the initial training set in active learning. 2010.
- [10] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.
- [11] Georg Kreml, Daniel Kottke, and Myra Spiliopoulou. Probabilistic active learning: Towards combining versatility, optimality and efficiency. In *Discovery Science*, pages 168–179. Springer, 2014.
- [12] Georg Kreml, Daniel Kottke, and Vincent Lemaire. Optimised probabilistic active learning (opal). *Machine Learning*, 100(2-3):449–476, 2015.
- [13] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168, 2011.
- [14] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Who should label what? instance allocation in multiple expert active learning. In *SDM*, pages 176–187. SIAM, 2011.
- [15] James F Bowring, James M Rehg, and Mary Jean Harrold. Active learning for automatic classification of software behavior. In *ACM SIGSOFT Software Engineering Notes*, volume 29, pages 195–205. ACM, 2004.
- [16] Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. Active learning in recommender systems. In *Recommender Systems Handbook*, pages 809–846. Springer, 2015.
- [17] Dilek Hakkani-Tur, Giuseppe Riccardi, and Allen Gorin. Active learning for automatic speech recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3904. IEEE, 2002.

- 
- [18] Min Tang, Xiaoqiang Luo, and Salim Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 120–127. Association for Computational Linguistics, 2002.
- [19] Michael I Mandel, Graham E Poliner, and Daniel PW Ellis. Support vector machine active learning for music retrieval. *Multimedia systems*, 12(1):3–13, 2006.
- [20] Dan Pelleg and Andrew W Moore. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2004.
- [21] Xiao Li, Da Kuang, and Charles X Ling. Active learning for hierarchical text classification. In *Advances in Knowledge Discovery and Data Mining*, pages 14–25. Springer, 2012.
- [22] Xiao Li, Charles X Ling, and Huaimin Wang. Effective top-down active learning for hierarchical text classification. In *Advances in knowledge discovery and data mining*, pages 233–244. Springer, 2013.
- [23] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.
- [24] JHU HLTCOE. Semeval-2013 task 2: Sentiment analysis in twitter. *Atlanta, Georgia, USA*, page 312, 2013.
- [25] Sara Rosenthal, Alan Ritter, Veselin Stoyanov, Svetlana Kiritchenko, Saif Mohammad, and Preslav Nakov. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, 2015.
- [26] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. Performance measurement framework for hierarchical text classification. *Journal of the American Society for Information Science and Technology*, 54(11):1014–1028, 2003.