A Data Mining Approach to Ontology Learning for Automatic Content-related Question-Answering in MOOCs

By:

SAFWAN MAHMOOD IBRAHIM
SHATNAWI

PhD 2016

Robert Gordon University

Doctoral Thesis

A Data Mining Approach to Ontology Learning for Automatic Content-related Question-Answering in MOOCs

By:

SAFWAN MAHMOOD IBRAHIM SHATNAWI

A thesis submitted in partial fulfilment of the

requirements of the

Robert Gordon University

for the degree of Doctor of Philosophy

OCTOBER 2016

DECLARATION

DECLARATION

I declare that all of the work in this thesis was conducted by the author except where otherwise indicated, and this thesis has not been submitted at any other university. Parts of the work outlined in this thesis have appeared in the following publications:

- Shatnawi, S., Gaber, M and Cocea, M. iMOOC: An Intelligent MOOCs Feedback Management System Architecture. The 8th European Conference on Technology Enhanced Learning, EC-TEL 2013.

-Shatnawi, S., Gaber, M. M., and Cocea, M. (2014). Automatic content related feedback for moocs based on course domain ontology. In IDEAL, Lecture Notes in Computer Science. Springer.

- Safwan Shatnawi , Mohamad Medhat Gaber , Mihaela Cocea. Text stream mining for Massive Open Online Courses: review and perspectives. In Systems Science & Control Engineering Vol. 2, Iss. 1, 2014.

Under revision: -Safwan Shatnawi , Mohamad Medhat Gaber , Mihaela Cocea. A Frequent Pattern Mining Approach to Developing a Subject Course Ontology: An Application to Question-Answering in MOOCs. In IEEE transaction on Learning Technology 2016. (Revised version submitted for the second round)

Safwan Mahmood Shatnawi.

ROBERT GORDON UNIVERSITY

Doctor of Philosophy

Abstract

A Data Mining Approach to Ontology Learning for Automatic Content-related Question-Answering in MOOCs

by SAFWAN MAHMOOD IBRAHIM SHATNAWI

The advent of Massive Open Online Courses (MOOCs) allows massive volume of registrants to enrol in these MOOCs. This research aims to offer MOOCs registrants with automatic content related feedback to fulfil their cognitive needs. A framework is proposed which consists of three modules which are the subject ontology learning module, the short text classification module, and the question answering module. Unlike previous research, to identify relevant concepts for ontology learning a regular expression parser approach is used. Also, the relevant concepts are extracted from unstructured documents. To build the concept hierarchy, a frequent pattern mining approach is used which is guided by a heuristic function to ensure that sibling concepts are at the same level in the hierarchy. As this process does not require specific lexical or syntactic information, it can be applied to any subject. To validate the approach, the resulting ontology is used in a question-answering system which analyses students' content-related questions and generates answers for them. Textbook end of chapter questions/answers are used to validate the question-answering

system. The resulting ontology is compared vs. the use of *Text2Onto* for the question-answering system, and it achieved favourable results. Finally, different indexing approaches based on a subject's ontology are investigated when classifying short text in MOOCs forum discussion data; the investigated indexing approaches are: unigram-based, concept-based and hierarchical concept indexing. The experimental results show that the ontology-based feature indexing approaches outperform the unigram-based indexing approach. Experiments are done in binary classification and multiple labels classification settings . The results are consistent and show that hierarchical concept indexing outperforms both concept-based and unigram-based indexing. The BAGGING and random forests classifiers achieved the best result among the tested classifiers.

Keywords: Data mining, ontology learning, question answering system, MOOCs, short text classification, frequent-pattern mining, association rule mining.

Acknowledgements

I would like to express my special appreciation and thanks to my supervisors Dr. Mohamed and Dr. Mihaela, you have been tremendous mentors for me. I would like to thank you for supporting and encouraging my research and for allowing me to grow as a research scientist. Your advice and support have been priceless.

Special thanks to my parents, my wife Amani, my daughters Raseel and Remas, and my son Hatem. Words cannot express how grateful I am for all of the sacrifices that you've made. Your prayers for me were what sustained me thus far. I would also to thank Dr. Amer Al-Badarnah and Dr. Emad Al-Shawakfa who took me to the realm of the scientific research and for all co-authors who honed my research skills. Special thanks to all teachers and professors who taught me through all my academic journey. At the end, I would also like to thank all of my friends who supported and incented me to strive towards my goals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

The advent of Massive Open Online Courses (MOOC) to the higher education field brought new opportunities for educational researchers to explore the teaching/learning processes and to better understand these processes. These opportunities arose due to the massiveness feature of MOOCs. The lack of any prerequisites to register in a MOOC allows a large number of registrants to participate in these courses. Those registrants have different purposes, academic backgrounds, ages, and languages [Ramesh et al., 2014].

A salient feature of MOOCs is discussion forums. Discussion forums allow MOOCs registrants to collaborate with their peers or course facilitators. Mainly, these discussions aim to fulfil cognitive needs of those registrants [Kanuka and Garrison, 2004]. However, in practise, they use it for different purposes beyond the cognitive needs [Ramesh et al., 2014]. MOOCs forums belong to the computer supported collaborative learning (CSCL) category. It is an important pedagogical element in MOOCs settings. Although MOOCs settings are

replicating online the stand-and-lecture pedagogies of conventional classrooms on a massive scale, it doesn't scale discussion sections. Course facilitators are not often capable of replying to all learners comments/questions.

The emergence of MOOCs have led to a controversial issue among educators in the higher education field. Some educators criticised it [Vardi, 2012, Laurillard, 2014, Konnikova, 2014]. Others considered it a solution for many problems in the traditional education settings [Kop et al., 2011, Coll et al., 2014]. Regardless of this debate among educators it brought new challenges for computer science researchers. Researchers investigate to what extent the existing techniques can support the new phenomenon. Can the existing techniques be used in new paradigms to allow personalised feedback to learners? What are the required features to overcome or mitigate the existing problems in MOOCs?

1.2 Motivation

MOOCs succeed in bringing hundred of thousands of learners to register in these courses [Coursera, 2013]. However, it failed in keeping them up to the finish line. A salient phenomenon in all MOOCs is the high drop-out ratio. The completion rate for most courses is below 13% [Onah et al., 2014a]. Research studies affirm that registrants had lack of support from their peers or course facilitators and they got lost after few lessons [Laurillard, 2014, Konnikova, 2014]. The current MOOCs settings don't scale up discussion forums to support the massive figure of registrants. This phenomenon gives the motivation for the research in this thesis. We aim to scale up the discussion forums to cope with the great number of MOOCs registrants. Specifically, we aim to offer automatic content related answers to students to fulfil their cognitive needs.

1.3 Research Questions, Aims and Objectives

This dissertation aims to automatically offer answers to content-related questions that appear in MOOCs discussion forums which are known to greatly contribute to the success of MOOCs. In order to achieve that goal we have outlined the following objectives for the research described in this dissertation:

- Develop a subject ontology from textual learning objects.
- Filter MOOCs discussion forums to identify content-related questions.
- Automatically answer content-related questions in MOOCs settings.

In order to achieve the aforementioned objectives we aim to answer the following questions:

- How can we represent a subject knowledge to support automatic content-related question answering systems? A subject content is found in different forms such as textbooks, slide notes, videos, transcripts, Blogs, and Wikies. However, these forms are not suitable for supporting automatic content-related question answering systems . In Chapter 5, we present a subject-content representation to support an automatic answering system. The content representation is not the final goal of this research. However, it is a key component for supporting discussion forums analysis module and the automatic question answering module.
- How can knowledge representations support MOOCs discussion forums analysis? MOOCs registrants use discussion forums for different purposes [Ramesh et al., 2014]. The first step in the automatic question answering module is to filter MOOCs discussion forums. In this step, we identify content-related questions. We process this task as a text classification

problem. MOOCs discussion forums data fall in the short text category. Short text classification problem has its own challenges. So, we propose an ontology-driven text classification approach to filter content-related posts in order to process and offer automatic feedback for these posts.

- How can the knowledge representation support automatic question answering systems and improve the quality of the returned answers? The current MOOCs settings and tools for managing discussion forums don't support registrants cognitive needs. A large number of registrants ask questions and send comments which contribute to the information overloading problem. A few questions are answered by course facilitators or other students which cause large portions of MOOCs registrants to quickly leave these MOOCs– this problem is known as "drop-out" problem. Although this is not the only reason for registrants to leave these courses, offering an automatic content-related answers for registrant questions helps learners to fulfil their cognitive needs and contributes to the solution for the high drop-out ratio problem. This module is able to answer questions belong to the "Remember, Understand, Apply, and Analyse" categories of Blooms's taxonomy [Bloom, 1956]. However, the other two higher level categories which are the "Synthesis and Evaluation" are not targeted in this thesis.

1.4 Contributions to The Knowledge

The work mentioned in this dissertation is a multidisciplinary research. It mainly targets data mining, ontology learning, and question answering systems for Education. As a result, our contributions are distributed over the aforementioned areas. As a result of the research which is described in this dissertation we have the following contributions distributed over the research areas:

- Massive Open Online Courses (MOOCs)
  - Our first contribution is a systematic review of MOOCs: MOOCs came with a hype in the media. Many educators wrote Blogs and posts to describe, to criticise, or to appraise MOOCs. However, there was a lack of scientific research studies that describe the MOOCs. Our first contribution is a systematic review of MOOCs to identify possible tools and techniques that contribute to the success of the new phenomenon.
- Ontology learning
  - Our main contribution is the representation of subject contents in a form of ontology. We proposed an approach for learning a subject ontology from the textual subject-content resources. We used different overlapped resources such as textbooks, slide notes, transcripts, Blogs, and Wikis to capture the subject concepts and their relationships. Then, we customised the FP-Tree structure using a heuristic function based on the FP-growth algorithm to build the subject concept hierarchy. We proposed an automatic natural language Deterministic Finite Automata (DFA) builder module for the extracted subject-concepts. We used this module to capture the concepts in the subject resources during the ontology learning process. And we also used it to capture subject-concepts in learners comments/questions.
  - Concept hierarchy construction: We used and proposed a customised version of the FP-tree algorithms to construct concept hierarchy for subject ontologies. We enhanced the resulting concept hierarchy using a heuristic function which is derived from the FP growth algorithm. To the best of our knowledge, we are the first to use these algorithms to construct the concept hierarchy for ontologies.

- Question Answering Systems for MOOCs
  - Automatic content related answering systems: A subject knowledge representation in form of ontologies allows semantic reasoning to answer content related questions. The question answering module takes the learners' questions as input. Then, it identifies the content related concepts and their properties. Next, it represents these concepts and properties as ontology triples patterns. After that, it queries the subject ontology to retrieve a proper feedback, i.e. an answer to the content-related question. Finally, it sends back the answer to the learners. This module aims to fulfil the cognitive needs for MOOCs registrants. To the best of our knowledge, subject ontologies have not been used to support question answering systems for educational purposes.
- Data Mining
  - Semantic feature indexing for short text classification: We investigate the impact of using different indexing approaches based on subject ontology when classifying short text in MOOCs forum discussions data. In this research, we use state of the art classifiers to measure what effects do the tested indexing approaches have on short text classification problem. We run the experiments in binary classification (content-related and non-content related) and multi-classes (general comment/question, general answer, content-related question, content-related answer) classification settings. The results are consistent and indicate that hierarchical concept indexing outperforms both concept indexing and unigram term indexing. The BAGGING and random forests classifiers achieved the best result among the tested classifiers. Chapter 7 discusses this contribution in details.

– Feature indexing We proposed two novel indexing approaches for short text classification which are concept-based indexing and hierarchical concept indexing. Both of these indexing approaches improves the accuracy of the state of the art classifiers. Although we tested these indexing approaches on MOOCs discussion forums, they can be extended to other short text domains.

1.5 Thesis Outline

Chapter 2 provides background and the main definitions for the areas related to this research including data mining, educational data mining, text mining, question answering systems, and ontologies. Chapter 3 reviews the state of the arts in areas important to the research in this dissertation. In Chapter 4, we outline the research methodology and the proposed framework for building a subject ontology to support the short text classification problem and the proposed question answering system. Chapter 5, describes the proposed module of the subject ontology learning from textual learning objects. It presents the proposed framework, algorithms, experimental work and analysis, and validation of the proposed framework. In Chapter 6, we present the short text classification module. It includes the data collection phase, the proposed feature indexing approaches, the experimental works and the results and analysis sections. Then, we present the proposed question answering system that utilises the resulting subject ontology in Chapter 7. Finally, Chapter 8 contains a summary of the research in this dissertation and highlights future directions for this research.

Chapter 2

Background

2.1 Introduction

The research in this dissertation covers different research areas including data mining, ontologies, and question-answering systems. The research primarily targets the educational domain. We used or customised techniques in different areas within the data mining field such as classification (short text classification), frequent patterns mining (FP-tree and FP growth), and topic detection. As a result, in this chapter we give a comprehensive background for data mining. On the other hand, Ontologies form the main component of the research in this dissertation. We proposed a subject ontology learning approach to represent a subject knowledge. The main resource for learning the subject ontology are the subject textual learning objects. As a result, we present a background for the ontologies and the ontology learning. The final component of this research is the question answering system that automatically answers content-related questions. So, we investigate the question answering systems and we highlight the basic components of a question answering system. All the research components target the new phenomenon in the higher education which is the Massive Open

Online Courses (MOOCs). We introduce MOOCs and its opportunities and limitations. Although these research areas seem to be disconnected, we integrate them to propose an automatic content related question answering system for MOOCs setting. The ontology learning module uses data mining techniques to build a subject ontology. Then this ontology is used to support short text classification to filter MOOCs discussion forums. The question answering system takes the output of the short text classification module and leverages the subject ontology to answer content-related questions in an interconnected framework. This chapter aims to give readers a comprehensive background of the aforementioned areas before delving to the framework details.

2.2 Data Mining

Evolution in computer hardware and software increases the amount of generated and stored data. This unbridled growth of data creates a need to reveal common patterns in daily businesses and scientific activities. Statistical and machine learning techniques have been used to learn and to discover hidden patterns in stored datasets. As a result, data mining field was emerged and flourished.

Data mining is defined as automatic or semiautomatic analysis of substantial quantities of data stored in databases, text documents, or images to discover reasonable, valid, and useful patterns. These patterns allow nontrivial prediction on unseen data [Liu, 2007, Witten and Frank, 2005]. The most common tasks of data mining are classification, topic modelling, clustering, association rule mining, and sequential pattern mining [Liu, 2007].

Traditional data mining uses structured data found in relational databases, spread sheets, or structured text files. However, due to the staggering volume of text documents and web pages, researchers focused on applying traditional data mining techniques to web and text documents. As a result, web mining and

text mining fields emerged. Unlike traditional data mining, web mining and text mining deal with unstructured, heterogeneous, or semi structured data [Liu, 2007].

Advent of web forums, Blogs, and social network sites like Facebook, MySpace, and Twitter allow users to interact with these sites and to send comments or feedback. A great number of users interact with these systems. As a result, they generate great volumes of continuous streaming data. Thus, ample of research studies focused on stream mining and social network analysis and mining. In stream data great volumes of continuous structured and unstructured data arrive at high speed and require real time analysis [Gaber et al., 2005, Aggarwal, 2011]. Data stream processing has its own challenges such as limited amount of memory and access to data points occurred in the order they arrived i.e. random access to the data points is not allowed [O'Callaghan et al., 2002]. In this research, that is described through the next chapters, we used data mining techniques to learn subject ontologies from subject learning units and to enhance the quality of the generated ontologies.

2.3 Educational Data Mining

A significant number of e-learning systems do exist on the Internet. Data mining and text mining techniques support these systems and scaffold its services. As a result, a new domain for data mining emerged and is known as Educational Data Mining (EDM). EDM aims to provide better experiences to learners when they interact with these systems. The advent of e-learning 2.0 creates new challenges for EDM. It adopted social learning via social software such as Blogs, forums, Wikis,etc. These systems allow learners to engage in the teaching process; moreover it allows learners to participate in peer grading which adds more challenges to the credibility of these systems. In order to motivate learners, to

keep them engaged, and to maximise learning these systems strive to personalise learner experiences. As a result, learner behaviours are analysed to deeply understand learners and to enhance the learning process which are the main objectives of Educational Data Mining (EDM). In this dissertation, we used data mining techniques for e-learning systems to classify discussion forums in MOOCs settings. We tested different state of the art classifiers to identify content-related questions in MOOCs discussion forums. Also, we tested two novel indexing approaches for classifying short text documents.

2.4 Text Mining

As defined earlier data mining aims to discover valid and useful information which allows nontrivial prediction. Structured data can be easily mined, however unstructured data mining such as text documents or stream text needs more intensive work before one could mine it. Many techniques were introduced to mine text data which are:information extraction, text summarisation, supervised learning, unsupervised learning, dimensionality reduction, transfer learning, probabilistic data mining techniques, cross-lingual mining, and text stream mining. First we will introduce unsupervised learning. Unsupervised learning methods do not require any manual labelling of the training data which is a laborious-intensive work. Manual labelling of the training data is used in other algorithms such as supervised learning and information extraction algorithms. Clustering and topic modelling are the commonly used techniques in unsupervised learning methods [Aggarwal, 2012].

Clustering is the process of grouping data instances based on specific similarity criteria. Data instances are referred to as objects or data points also. [Liu, 2007]. Clustering methods preliminary were designed for quantitative and categorial data. However, some clustering algorithms such as K-means and K-medoid

were used to cluster text data later on. Native clustering methods do not work effectively for text data since text data have sparse, high dimensionality representation, and different text representation. Hence, text clustering requires a specific text clustering algorithms to handle text document representation issues. Feature selection is the first step in text mining. This process is crucial to the quality of text mining methods. Noisy features must be eliminated before delving into the clustering process. On the other hand it selects relevant features. Variant feature selection approaches were used in text mining such as frequency-based selection, term strength selection, term contribution, and entropy-based ranking. Another method in text preprocessing is feature transformation which aims to improve the quality of document representations. These methods include latent semantic indexing, Non-Matrix factorisation, and probabilistic latent semantic analysis [Aggarwal and Zhai, 2012a].

2.5 Text Clustering Algorithms

Text documents are clustered based on similarity. Different similarity functions have been used in text clustering. A popular similarity function is cosine similarity. Also, heuristic functions such as Term Frequency (TF), Inverse Document Frequency (IDF), and document length normalisation have been used to optimise similarity functions [Aggarwal and Zhai, 2012a]. Probabilistic models of text, represent text documents with probability distribution over words, it obtains similarity according to information theoretic measures [Zhai, 2008].

2.5.1 Agglomerated hierarchical algorithm

Agglomerated hierarchical algorithms were used extensively in clustering quantitative and categorical data. Then, it was found that is applicable to apply

these algorithms for text data. Agglomerated clustering algorithm starts with individual documents in the corpus as initial clusters, where each document represents a cluster. Then, it iteratively merges similar documents to form new higher layer clusters. And finally it ends up with the trivial cluster consists of all documents in the corpus. According to [Murtagh and Contreras, 2012] hierarchical algorithms fall in three categories which are linkage methods and centroid, median, and minimum variance methods.

Hierarchical linkage based methods practise one of the following three similarity approaches:

• Two groups of clusters are merged if it has least interconnecting dissimilarity among all other documents pairs which is called *single linkage clustering*. It is extremely efficient method for clustering text document. However it suffers a drawback of chaining phenomenon in which incompatible documents are group in the same cluster. As a result generate poor quality clusters.

• Instead of clustering document based on the maximum similarity among documents pairs. Clusters are obtained by computing the average similarity of all possible combinations of documents pairs of the clusters, a method known as *group-average linkage clustering*. The more documents in the clusters the less efficiency of this method. However, it generates good quality clusters.

• Two groups of clusters are merged based on the worst-case similarity between two pairs of documents. Although this method overrides the chaining phenomenon exists in single linkage clustering method, it is computationally more expensive than the aforementioned linkage methods; this method is known as *complete linkage clustering*.

2.5.2 Partitional Clustering Algorithms

Partitional clustering methods create flat(one level) partitioning of the data points (text documents). These methods find all desired clusters at once. K-means and K-medoid are the most two algorithms used with text data, the former starts with set of kernels documents not necessarily from the original corpus. Each of these document is used to build the cluster around by assigning documents in the corpus to one of these kernels using closest similarity. In the next iteration the original kernel is replaced by the centroid of the assigned documents. The algorithm is terminated when convergence is achieved. K-medoid selects kernels from the original documents in the corpus. Then, it builds clusters around these kernels. Each document is assigned to the closest kernel using average similarity of the document to these kernels. Iteratively the algorithm improves the kernels using randomise interchanges. It uses an objective function to determine whether the interchanges process improves the cluster or not in each iteration. Once a convergence is achieved the algorithm is finished.

Performance wise k-means outperforms k-medoids and generates better quality clusters than k-medoids, this is because k-means requires few iteration to converge. On the other hand k-medoids inefficiently works when it is applied to sparse data [Aggarwal and Zhai, 2012a]. A variation of k-means algorithm also was used with text document which is 'bisecting' k-means. A comparison study found that bisecting k-means outperforms the original k-means algorithm and as good as or better than agglomerated clustering algorithms for variant evaluation measures [Steinbach et al., 2000].

2.5.3 Hybrid Text Clustering

Hierarchical clustering algorithms are not efficient since it is computationally expensive. However, it generates robust clusters. In contrast partitional algorithms are computationally efficient. Nevertheless they are not effective in terms of quality of the generated clusters . Many attempts were introduced to improve both efficiency and effectiveness of text clustering algorithms. The initial selection of the seeds for k-means algorithm significantly contributes to the quality of the generated clusters. As a result, many hybrid algorithms [Luo et al., 2009,Cutting et al., 1992] attempted to find good initial seeds for k-means algorithm. Others [Cutting et al., 1992] proposed algorithms to refine cluster centroid, claiming that this refinement enhances the effectiveness of the generated clusters. In this section we introduce the most significantly recognised improvements.

The proposed clustering algorithm in [Cutting et al., 1992] starts by finding good initial seeds for the k-means algorithm. This is achieved by implementing two methods which are *buckshot and fractionation* as they called in [Cutting et al., 1992]. The former randomly selects $\sqrt{kn}$ documents, where k is the number of desired clusters and n is the number of documents in the corpus. Next, an agglomerated algorithm is used to cluster this sub group into k clusters, where the centroid of each cluster forms a seed for k-means algorithm. It is important to mention that multiple runs of this algorithm against the same corpus will not generate the same partitions. However, in practise [Cutting et al., 1992] found multiple runs gave qualitatively similar partitions. The latter brakes the corpus into fixed size n/m, *where* $m > k$. Next, an agglomerated algorithm produces z clusters for each group. As a result, it generates zm clusters. Each cluster is considered as an individual document by merging all documents in that cluster. This process is repeated till it obtains k clusters. The obtained k clusters form the seeds for k-means algorithm. Then, every document is assigned to the nearest cluster. As a result, it modifies the cluster

centroid to include the new document. So, the new centroid replaces the old one and is used as a seed in the next iteration.

2.5.4 Frequent Term-Based Text Clustering

One of the main challenges for text clustering is the large dimensionality of the document vector space. Frequent term-based clustering methods group documents based on subset of the frequent terms set, instead of the whole terms in the collection. It obtains the frequent item set using the association rule mining method. Many algorithms were introduced to find the frequent items set that has minimum support, more details in [Agrawal and Srikant, 1994,Han et al., 2000, Zaki, 2000]. Frequent term-based clustering algorithms consider each selected subset of frequent terms set to represent a cluster description. On the other hand, those documents that cover all subsets of these frequent terms set to represent the cluster itself.

The work in [Beil et al., 2002] presented two algorithms for text clustering based on frequent terms set which are: frequent term-based clustering (FTC) and hierarchical frequent term-based clustering (HFTC). The former is bottom-up flat clustering algorithm starts with an empty set of clusters. Then, In every iteration it selects one of the cluster description (one set of frequent-term sets) that has minimum overlap with other clusters. The selected set will be removed from the database and the documents cover this set also removed from the document collection. The algorithm ends when all documents in the collection are clustered. It generates clusters with no overlap. The latter algorithm exploits the monotonicity property of frequent item set where all k-1 subset of frequent k-terms are also frequent. It starts with one big cluster contains all documents. Then, in next iterations, it clusters the document based on frequent 1- term set. Consequently, it uses 2-terms sets. And continue until no more frequent k-terms exist. Clusters generated by this algorithm are overlapped.

2.5.5 Graph Based Text Clustering

Using graph model for clustering back to 1959 [Augustson and Minker, 1970], where the maximum complete subgraph of a graph was defined as cluster. In [Dhillon, 2001] a method to cluster text documents and words also known as co-clustering was introduced based on bipartite graph structure. Documents and words represent vertices, E is set of edges between documents and words. In this structure no edges between words itself nor documents itself, i.e only document to words edges exist. Edges are positively weighted. The weights represent the word frequency in a document. To cluster document the cut function is defined to partition vertex set V $cut(v_1, v_2) = \sum_{i \in v_1, j \in v_2} M_{ij}$. Finding minimum cut in vertices set V is an NP complete problem. However, heuristic methods are used to find minimum cut. Spectral graph bipartitioning is used. As a result, V is partitioned to nearly equally sized two subsets V_1^*, V_2^* and this will give the document clusters. Word clustering is obtained by assigning words to the greatest edge weight connected document and simultaneously performs k-means algorithm to obtain the bipartition.

Another graph based approach introduced in [Aslam et al., 2006] where they represented the documents in the corpus using similarity graph G. The cosine similarity between documents is calculated and an weighted edges between document set D is E. E represents the edges between $d_i, d_j \in D$ the weight of each $e \in E$ represents the similarity value. Unlike the work in [Dhillon, 2001] where edges exists between words and documents only, in [Aslam et al., 2006] edges exist between documents only. Similarity ratio σ is set and represent the minimum threshold where all edges under σ are ignored. Given G_σ subgraph the highest similarity edge is set as the centre of the cluster (star as called in their work). All connected vertices(satellites) to this star form a cluster. Similarity of satellites and star is guaranteed. However, similarities between satellites are not guaranteed. Theorems exist in this method failed to prove similarities among

satellites. However, they claimed that experimental results prove similarities among satellites.

Neighbours-based clustering algorithm is also a graph based algorithm proposed in [Luo et al., 2009] to select good well separated initial seeds for k-means algorithm based on pairwise similarity value, link function value, and number of neighbours of documents in the corpus. It uses new similarity function for assigning documents to the nearest centroid. Finally a heuristic function selects the candidate cluster to be split for bisecting k-means.

The first step in this algorithm is finding similarities between (d_i, d_j) for all document pairs in the corpus using cosine similarity. If the similarity value above given θ specified by the user, then the pairs of the documents (d_i, d_j) are considered neighbours. The similarity information are represented using nxn matrix M, where n is the number of documents in the corpus. Each value in this matrix is represented using binary representation where 1 in M[i,j] means documents $d_i and d_j$ are neighbours and 0 otherwise. The number of neighbours for document d_i denoted by N(d_i) is $\sum_{j=1}^{n} M[i,j]$. The second function is link function of documents pairs $(d_i and d_j)$ which is the number of common neighbours between $d_i and d_j$. They calculate the value of the link function by multiplying the i^{th} row by the j^{th} col which is denoted by $link(d_i, d_j) = \sum_{m=1}^{n} M[i,m] \cdot M[m,j]$. The value of this function is proportionally related to the probability of $d_i and d_j$ belong to the same cluster. Next the algorithm find candidates seeds for the k-means algorithm by selecting (k+p) documents as candidates seeds set Sc, where k is the number of desired seeds and p any extra number of documents specified by the user. The set of candidate seeds are selected from the first minimum (k+p) N(d_i) value documents. After that the algorithm finds similarity and link values for all documents pairs combinations in Sc. Based on these values it calculates the $rank_{cos}$ and $rank_{link}$ for every pairs of documents in Sc. The sum of the $rank_{cos}$ and $rank_{link}$ gives the $rank_{total}$ value.

2.5.6 Other Clustering Methods

-Winnowing-Based Text Clustering: winnowing algorithm was introduced by [Schleimer, 2003] to find similar text across documents to detect copy and past or plagiarism in research and student papers. The algorithm divides the document into k-gram substring where k value is specified by the user. Each k-substring is called hash. Some subset of these hashes will be selected to represent the document fingerprint. When two or more documents share one or more fingerprints they are considered similar. Based on that [Parapar and Barreiro, 2008] proposed text clustering algorithm. Experimental results show that winnowing based text clustering outperforms k-mean and term frequency representations.

Table 2.1 shows text clustering algorithms summary. The table contains brief description of every clustering algorithm mentioned in the review. Also the category of these algorithms and its limitations and computational complexities. Text clustering algorithms have many categories including hierarchical, partitioning, graph based, hybrid, and frequent item. Hierarchical algorithms suffer from the chaining phenomenon. However, overcoming this problem is possible with high computational cost. Partitioning algorithms generate robust clusters. Also, they are computationally efficient.

Text clustering algorithms can be used to build subject concept hierarchy for subject ontologies. We proposed a novel approach similar to the frequent-terms based text clustering to create subject concept-hierarchy. We considered each learning unit a data point. On the other hand, each frequent concept a cluster description.

TABLE 2.1: Text clustering algorithms summary

Algorithm	Category	Description	Limitation and Computational efficiency
Single Linkage	Hierarchical	Merge data points based on least interconnect dissimilarity	Chaining phenomenon, computationally efficient. Time complexity O(NlogN). Space Complixity O(N).
Group Average Linkage	Hierarchical	Merges data points based on the average similarity of all possible combination of documents	Overcome chaining phenomenon. Computationally expensive O(N^3). Space O(N^2).
Complete Linkage	Hierarchical	Merges data points based on the worst similarity	Overcome chaining phenomenon. Computationally expensive O(N^2). Space O(N).
K-means	Partitioning	Starts by set of kernels documents not necessarily from the original corpus and build the clusters around these documents using closest similarity. The centroid of the cluster is used in next iteration. Bisecting k-means is a variation of k-means	Computationally efficient O(N log N). Requires few iterations to converge. Outperforms the K-medoid
K-medoid	Partitioning	Starts by set of kernels documents from the original corpus and build the clusters around these documents using average similarity. The quality of the clusters is improved using objective function.	Robust clusters generated
Bipartite graph	Graph based	Documents and words are represented as bipartite graph. Cut function is used to cluster documents	NP complete problem. Heuristic function used for optimal solution.
Buckshot and fractionation	Hybrid	k-means based clustering method. With improvement in the kernels set selection using buckshot and fractionation	Multiple runs of this algorithm generates different clusters.
FTC and HFTC	Frequent item	Reduce the dimensionality of documents by representing document using its frequent terms set. FTC is bottom up clustering. HFTC is top down clustering	FTC generates clusters without overlapping. HFTC generates overlapped clusters. Both FTC and HFTC outperforms K-means and K-medoid.
Star-Satellites	Graph based	Data points are represented as similarity graph. Cosine similarity function are used. A star is the centre of the cluster. documents that are above user defined threshold similarity are satellites	Similarity between satellites are not guaranteed. Theorem exist in this method failed to prove satellites similarities.Time complexity O($Nlog^2N$),.
Windowing- Based	Partitioning	It divides the document into k-gram sub strings. Then subset of these sub strings represent the document fingerprints. Documents share two or more fingerprints considered similar and clustered.	Outperforms K-means and FTC.

2.6 Topic Modelling

One of text clustering challenges is the large volume of words (terms) in documents. Many methods emerged to reduce the large volume of the documents by representing documents using a small subset of their words. These words represent the abstract (theme) of the documents. Researchers used statistical modelling to abstract text documents. Statistical modelling is known as topic

modelling in machine learning and natural language processing [Blei, 2012,Landauer et al., 2007]. Statistical modelling for topic detection and tracking includes but is not limited to: Latent Semantic Indexing and Latent Dirichlet Allocation.

2.6.1 Probabilistic Latent Semantic Analysis (PLSA)

The vector space model that is used to represent documents is a high dimensional sparsely space. Latent Semantic Analysis (LSA), also called Latent Semantic Indexing (LSI), is an automatic indexing method. It projects documents and words into a lower dimensional space. The projected terms represent the semantic concepts in these documents which hopefully overcome synonyms and polysemy problems where different terms have the same meaning or a term may have different meaning according to the context. This projection makes it possible to analyse documents at conceptual level. LSA has its root in the information retrieval field which is used for indexing information retrieval system. LSA uses Singular Value Decomposition (SVD) to project documents and words into k-latent semantic spaces. Similarity between documents is measured using the latent semantic space and so are the word similarities. More details about LSA can be found in [Aggarwal, 2012].

Typically, documents are added to the collection (corpus) rapidly. As a result, the document-term matrix needs updating, which in turn, leads to re-calculation of the latent semantic space to reflect the new added documents. Repeating the whole process is computationally inefficient. Instead, two methods were used which are fold-in and semantic space updating methods. The former computes the projection of the new documents using the existing latent semantic indexing, which is computationally efficient. The latter overcomes the outdated models by adding new documents to the collection over time, however, indexing is not guaranteed to provide the best rank approximation. Probabilistic LSA model

was introduced by [Hofmann, 1999]; this approach aims to statistically model co-occurrence information by applying a probabilistic framework to discover the latent semantic structure. The latent variables (topics) are associated with observed documents [Hofmann, 1999].

2.6.2 Latent Dirichlet Allocation (LDA)

LDA is a generative probabilistic model. In practice, documents contain multiple topics and words distributed over many topics. LDA model aims to capture all topics in these documents. It considers a topic as a distribution over words. These topics are generated in advance. For each document LDA draw some topics that cover that document. Then, a topic is assigned to each word in the document and a word is selected from the topic words distribution. In practice, topics, document topics distribution, and document words distribution over topic are unknown or hidden. Only documents are observed. As a result, the computational problem for topic modelling is to infer all hidden structures given the observed document. A document collection of scientific research journals from 1880 to 2002 was used to test the LDA model. These documents were not labelled and have no metadata attached to them, i.e. only text in these documents were observed. They assumed 100 different topics exist in these documents. And they used the LDA model to infer word distributions over these topics and topic distributions over all documents. Also, they studied how topics evolved over time [Blei et al., 2003].

2.6.3 Hierarchical Generative Probabilistic Model

The hierarchical generative probabilistic model (HGPM) based on bigram model was introduced in [Wallach, 2006]. Marginal and conditional word counts are obtained from a corpus. The marginal count is the number of times a word

occurred in the corpus. The conditional count is the number of times a word w_i immediately followed another word w_j. Unlike LDA where word positions are ignored, in this model each word w_k is predicted based on the word w_{k-1}. The bigram model based on the marginal and conditional counts predicts w_k given the observed w_{k-1}. However, this approach integrates bigram-based and topic-based models to achieve a better predictive accuracy over the LDA and hierarchical LDA models.

An extension to the bigram model was introduced by [Tam and Schultz, 2008]. They presented a correlated bigram LSA approach for unsupervised language model (LM) adaptation and used it for automatic speech recognition. They presented a technique for topics correlation modelling using Dirichlet-Tree prior. They proposed an algorithm for bigram LSA training via variational Bayes approach and model bootstrapping, which is scalable to large language model settings. Moreover, they formulated the fractional Kneser-Ney smoothing to generalise the original Kneser-Ney smoothing which supports only integral counts.

2.6.4 Discriminative Probabilistic Model

Topics in text documents evolve over time. Thus, studying the time factor for topic modelling is one of the factors that research studies examined. A research study examined the time factor in documents [He et al., 2010]. Instead of representing documents using the word vector space only, they represented documents using words and time vector spaces. They proposed a temporal discriminative probabilistic model for both offline and online topic detection and they evaluated it for performance issues. In addition, they investigated several types of topic detection models, namely, deterministic, discriminative and probabilistic mixture, and mixed membership. Experimental results showed that a simple deterministic mixture is more efficient and effective than sophisticated models such as the LDA model.

The discriminative probabilistic model estimates posterior (conditional) probability of a topic given an observed document. Adding a temporal element achieves best performance/complexity trade-off. In the offline topic detection model they assumed the existence of a set of features that discriminate documents in the corpus. They eliminated stop words and rare words from these features. The probability of a new document is obtained by computing the conditional probability of the new document for all sets of discriminative features. On the other hand, online topic detection incrementally examines each incoming document to assess whether it belongs to a new topic or an existing topic. Some researchers named this process as evolved topic detection instead of online topic detection [Aggarwal et al., 2003].

2.6.5 Non-Probabilistic Topic Detection

A non-probabilistic online topic detection model was introduced in [Allan et al., 2005] to cluster news stream. It detected events (topics) and assigned incoming stories to one of the existing topics or creating a new topic when incoming stories contains a new topic. It represented each document using the top 1000 weighted words that occur in a story as a vector, using the vector space model. Its similarity to every previous document is calculated using the cosine similarity function. It assigned the document to the nearest neighbour when the similarity value is above a given threshold or created a new topic when the similarity is below that threshold. The authors explored several techniques to enhance the quality of the topic clusters, such as different weightings for words, different criteria for document selection and penalties. These, however, did not lead to significant improvements in the quality of the resulting clusters. Finally, when they used the average-link clustering, where every cluster is represented by its centroid, the generated clusters were more robust and they achieved better computational efficiency. Table 2.2 presents a summary of topic detection methods.

TABLE 2.2: Topic Detection Techniques Summary

Approach	Category	Description
LSA	Vector space model	Using SVD to project documents and words into lower dimensional space. Fold-in and Semantic space updating are two methods to enhance computational efficiency.
PLSA	Probabilistic	Statistically model co-occurrences information using aspect model. It applies probabilistic framework to discover latent semantic structure.
LDA	Probabilistic	Generative probabilistic modelling aims to capture multiple topics exist in a document.
Hierarchical Generative model (Bigram)	Probabilistic	Extends LDA where word position and co-occurrence are considered. Bigram model based on marginal and conditional counts is used. Space complexity O(V^2K), V: vocabulary, K: topics
Discriminative Model	Probabilistic	Time factor is considered. Documents are represented by words and times vectors. The temporal discriminative probabilistic model is used for online and online documents.
Experimental Model	cluster-based	Clustering documents based on the topics exist in these documents. Experimentally many techniques were implemented. Average link clustering outperforms other used clustering techniques.

In this dissertation, we proposed an approach for topic modelling using subject ontologies. The aim of this module is to identify the topic of student posts in MOOCs settings. Since student posts and comments are relatively short text and require processing in real time settings, most of the aforementioned topic modelling approaches are not appropriate to MOOCs settings. As a result, we used subject ontology to identify topics in student posts. Then, we used subject concept hierarchy to label students post.

2.7 Text Classification

In data mining, the classification problem uses a set of training records (training set) to construct a classification model. Then, the model is used to assign a label to each unseen record in the test records (test set). Typically, classification models are used to predict categorical values. However, the regression modelling problem, which is a variation of the classification problem, assumes continuous values instead. On the other hand, text classification is an instance of the classification problem that uses a set-valued features as predictors. A document is represented as a bag of words disregarding grammar and word

order. Features (words in the corpus) are much greater than a traditional set-valued classification [Aggarwal and Zhai, 2012b]. Formally, text classification, is defined in the definition 1 [Sebastiani, 2002].

Definition 1. Text Classification TC
Let D be a set of documents $D = \{d_1, d_2, ..., d_n\}$
Let C be a set of predefined categories $C = \{c_1, c_2, ..., c_m\}$
Then, text classification (TC) is the task of assigning a Boolean value to each pair of $\langle d_j, c_i \rangle \in D \times$ C.
A function $F : D \times C \rightarrow$ {True, False} is called the classifier model.

2.7.1 Bayesian (Generative) Classifiers

Bayesian classifier is a probabilistic classifier. It is a conditional probability model for constructing classifiers. Each point (document) in the data space is represented as a vector of features. It assumes that these features are independent. Naive Bayesian models are the distribution of the documents in each class. A document is represented using "Bag of Words" as its features [Aggarwal and Zhai, 2012b].

2.7.2 Decision Trees

Decision trees are typically used as one of the inductive learning methods. Given a classification task, the classification rule is expressed as a decision tree. A decision tree requires features in the training set to provide sufficient information to differentiate between classes. Otherwise, it is impossible to develop a classification rule. Leaves of a decision tree are class labels, intermediate nodes represent attribute-based tests with a branch for each possible outcome. Branches do not necessarily have the full set of features. A branch may have a subset of the features and still can classify an object [Quinlan, 1986]. Their

robustness to noisy data and their capability to learn disjunctive expressions seem suitable for document classification. The work in [Li and Jain, 1998] used C5, a successor of the ID3 algorithm which was proposed in [Quinlan, 1986], to classify text documents.

2.7.3 Pattern (Rule)-based Classifiers

In fact decision trees is a rule based classification method. Each branch represents a rule. However, the decision tree framework is a strict hierarchical partitioning of the feature space. Rule-based classifiers model the feature space as a set of rules. Each rule is a condition on the underlying feature set. Each rule or subset of rules is mapped to a class. The set of rules must cover all the points in the decision space. The work in [Johnson et al., 2002] presented a decision-tree-based symbolic rule induction system for categorising text documents automatically. Their method for rule induction involves the novel combination of a fast decision tree induction algorithm introduced by [Quinlan, 1986] designed for text data. Also, they proposed a method for converting a decision tree to a simplified rule set which is logically equivalent to the original tree.

2.7.4 Support Vector Machines (SVM) Classifiers

As most of the text classifiers, the SVM classifiers were primarily proposed for numerical data [Cortes and Vapnik, 1995]. SVM non-linearly transforms input vectors to a high dimensional feature space. Then, a linear decision surface (hyper-plain)is constructed which can best separate the different classes. Only small amount of the training data (support vector) is used to construct this hyper-plain. SVM has a 30 years history from 1965 to 1995 [Vapnik and Kotz, 1982]. After that, SVM was used for text classification [Joachims, 1996].

2.7.5 Neural Network Classifiers

Neural networks introduced by [Rosenblatt, 1961] as a learning model similar to the perceptron model in human brains. It consists of an input layer with minimum two nodes and an output layer with an output node. The input node is connected to the output nodes using weighted connection. On the other hand, a typical neural network has several hidden layers. During the learning process these weights are adjusted to correctly predict the output. An advantage of neural networks model is its low computational expense. However, it can learn problems that are linearly separable. Neural network used for text classification first by [Ruiz and Srinivasan, 1998].

We used text classification algorithms for classifying MOOCs discussion forums. MOOCs registrants use these discussion forums for different purposes and play different rules. As a results, it is important to automatically label these discussion forums. In this dissertation, we used text classification algorithms to filter posts that contain content related questions. These posts are processed by a question answering system to answer registrants' questions. We proposed two indexing approaches for improving the accuracy of these classifiers. Chapter 6 presents our work for classifying MOOCs discussion forums.

2.8 Frequent Pattern Mining

The frequent pattern mining problem aims to find relationships among items in a transaction database. A frequent pattern should present in at least a fraction s of these transactions. This fraction is referred to as the minimum support. Formally, the frequent pattern mining is defined in Definition 2. The problem was first proposed in the context of market basket data to discover frequent groups of items that are bought together [Agrawal et al., 1993]. Another problem proposed in [Agrawal et al., 1993] was the association rule which is related

to the frequent pattern mining problem. Association rule problem aims to find associations between sets of items with some minimum specified confidence c and some minimum support s. Confidence is a value indicates how often a rule has been found to be true. Confidence is defined in equation 2.1. A rule such as $T_i \Rightarrow T_j$ is considered an association rule if T_i and T_j are frequent pattern with a confidence $>$ c for $T_i \cup T_j$. It is obvious that 0 $<$c $<$1. After that, a number of techniques have been proposed for frequent pattern mining. These techniques include Frequent Pattern Mining with the Traditional Support Framework, Interesting and Negative Frequent Patterns, Constrained Frequent Pattern Mining, and Compressed Representations of Frequent Patterns [Aggarwal and Han, 2014]. We will introduce one of these algorithms since we used it as a component in the ontology learning module.

$$CONF(T_i \Rightarrow T_j) = Support(T_i \cup T_j)/Support(T_i) \tag{2.1}$$

Definition 2. Frequent Pattern Mining
Let I ={ i_1, i_2, ..., i_k } be set of items.
Let D={ T_1, T_2,...,T_n} be a transaction database; where $T_i \subset$ I.
Let s be a minimum support value where i_t appears in $T_i \Leftrightarrow i_t > s$. Then, find all P $\subset$ I; where P $>$ s.

2.8.1 FP-Tree and FP-Growth

The FP-Growth Algorithm finds frequent patterns without using candidate generations, thus improving performance. It uses a divide-and-conquer strategy. It uses a special data structure called the frequent-pattern tree (FP-Tree). FP-tree is a compact structure that stores quantitative information about frequent patterns (cf. definition 2) in a transaction database; it stores items and their frequencies. The FP-tree represents the conditional transaction database T_i

with the use of compressed prefixes. It divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately [Han et al., 2000].

Definition 1. Frequent Pattern Tree (FP-Tree) is a tree structure defined as follows:

A It has one root node, a set of item-prefix subtrees as the children of the root, and a frequent-concept header table.

B Each node in the item-prefix subtrees consists of three fields:

1. item name: registers which item is represented by the node;
2. occurrence frequency: the number of transactions represented by the portion of the path reaching the node;
3. and node-link: refers to the next node in the FP-tree carrying the same item, or null if there is none.

C Each entry in the frequent-concept header table consists of two fields: (a) item name and (b) head of node-link, which points to the first node in the FP-tree carrying the item.

Algorithm 1 was proposed by [Han et al., 2000] to construct the FP-tree structure for a transactional database D. It takes a transaction database as input. First, it constructs the header table which is a data structure that contains all distinct items in the database along with their frequencies in descending order. It sorts items in each transaction according to the header table entries. Then, it process all transactions in the database to construct the FP-tree structure. Each node (item) in the header table contains a link to the first instant of that item in the FP-tree. When a new node for that item in created a link to the new node is created to connect that node to the previous created node. For each transaction, the algorithm starts from the root node and search for the item in

Algorithm 1 BuildFPTree (DB, θ)*

Input: A transaction database DB and a minimum support threshold θ.
Output: Its frequent pattern tree, FP-tree.

1: Scan the transaction database DB once. Collect the set of frequent items F and their supports. Sort F in support descending order as L, the list of frequent items.
2: Create the root of an FP-tree, T, and label it as "null". For each transaction Trans in DB do the following. Select and sort the frequent items in Trans according to the order of L.
3: Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list.
4: Call insertTree([p|P], T).

```
function INSERTTREE([p|P],T)
    if T has a child N then and N.item-name = p.item-name
        N.count ++
    else
        create a new node N.
        Let N.count ← 1.
        Let N.parentLink ← T.
        Let N.node-link ← the nodes with the same item-name via the node-
link structure.
    end if
    if P is nonempty then
        insertTree([p|P],N)
    end if
end function
```

* FP Tree algorithm as proposed by Han et al. 2000

that level. If a node for that item exist, then that item's count is incremented. Otherwise, it creates a new node for that item and set the item count to 1. and moves to the next level. Once it constructed the FP-Tree it is possible to mine it to find the complete set of frequent patterns. Han proposed the FP-Growth algorithm 0 to mine the resulting FP-tree [Han et al., 2000].

We customised the FP-tree algorithm to build concept hierarchy for subject ontologies. And then, we used the association rule mining to improve the quality

Algorithm 2 FPGrowth(DB,FP-Tree)*

Input: A database DB, represented by FP-tree constructed according to Algorithm 1, and a minimum support threshold θ.

Output: The complete set of frequent patterns.

call BuildFPTree(FP-tree, null).

procedure FP-GROWTH(Tree, a)

if T **then**ree contains a single prefix path

let P be the single prefix-path part of Tree;

let Q be the multipath part with the top branching node replaced by a null root;

for e **do**ach combination (denoted as β) of the nodes in the path P do

generate pattern $\beta \cup$ a with support = minimum support of nodes in β;

Let freq pattern set(P) be the set of patterns so generated;

end for

else

let Q be Tree;

for e **do**ach item ai in Q do

Generate pattern $\beta = a_i \cup$ a with support = a_i .support;

Construct β's conditional pattern-base and then β's conditional FP-tree Tree β;

if T **then**ree $\beta \neq \phi$

Call FP-growth(Tree β , β);

Let freq pattern set(Q) be the set of patterns so generated;

end if

end for

end if

Return (freq pattern set(P) $\cup$ freq pattern set(Q) $\cup$ (freq pattern set(P) $\times$ freq pattern)

set(Q) }

end procedure

* FP Growth algorithm as proposed by Han et al. 2000

of the resulting concept hierarchy. Customising the FP-tree structure to build concept hierarchies for ontologies is a novel approach in the web semantic field. We discuss our novel approach in Chapter 5.

2.9 Ontology Learning

An ontology is an explicit formal specification of a shared conceptualisation of a domain of interest [Studer and Staab, 2009]. An ontology defines the intentional part of the underlying domain, while the extensional parts of the domain (knowledge itself or instances) are called the ontology population. An ontology is formally defined in [Hotho et al., 2002]. Definition 2 formally defines an ontology.

Definition 2. A core ontology is a sign system $\Theta := (T, P, C^*, H, Root)$, where
T: a set of natural language terms of the Ontology
P: a set of properties
C^*: a function that connects terms t $\in T$ to a set $p \subset P$
H: a hierarchical organisation connecting all terms from T in a cyclic, transitive, directed relationships.
Root: the top level node where all concepts in C^* are mapped to it.

Developing an ontology is a knowledge engineering task [Hatala et al., 2012,Sure et al., 2006, Cimiano et al., 2009]. Developing and maintaining an ontology remains a costly and resource-intensive task. Therefore, techniques to support ontology development and to maintain existing ontologies are important to facilitate ontology adoption in different systems for different domains. These techniques aim to overcome the ontology development drawbacks. Also, for the educational field stakeholders, it is important to hide the complexity and the

technicality of an ontology developing. In the ontology development field, these supporting techniques are called ontology learning.

Ontology learning is concerned with knowledge acquisition. It consists of several phases which are: term extraction, synonyms, concept identification, concept hierarchy, relation identification, and sets of rules [Cimiano, 2006].

Figure 2.1 shows the general ontology learning layer cake [Buitelaar et al., 2005]. However, in ontology learning for education, specifically for subject course ontologies, we believe that the ontology layer cake should be reduced to the four middle layers which are: concepts, synonyms, concept hierarchy, and relations. Since we aim to explicitly represent a subject knowledge which is a special case of the general ontology learning. A subject course ontology should match the level of information found in a textbook on that subject. However, ontology-driven applications for educational purposes can add their own rules to achieve their functions.

A number of ontology learning researchers explored Natural Language processing (NLP) techniques to discover domain concepts and relationships among concepts [Valencia-García et al., 2004, Maynard et al., 2008]. Researchers used semantic similarity to support the ontology learning process. The work presented by [Chen et al., 2006] used Latent Semantic Analysis (LSA) to support the concept discovery. A recent research utilised the web page structure to discover the underlying concepts and properties of a domain [Ahmed et al., 2012]. They leveraged Wikipedia structure to retrieve concept definition and to identify existing relationships. Mining domain specific glossaries and texts to enrich and evaluate ontologies is proposed in [Parekh and Gwo, 2004]. Table 2.3 summarises some of the tools developed to build domain ontologies from text.

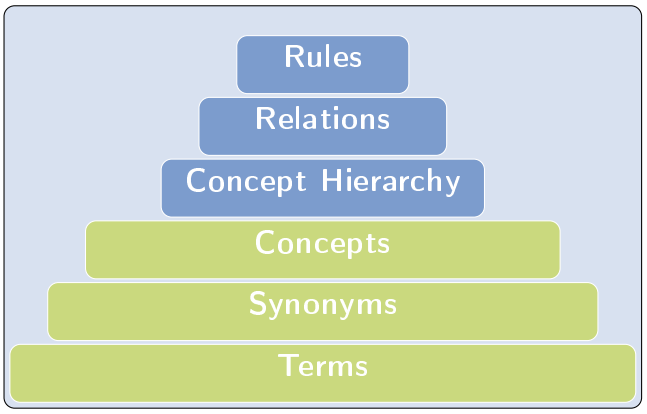


FIGURE 2.1: Ontology Learning Layer Cake as Proposed by [Buitelaar et al., 2005]

TABLE 2.3: Ontology Learning from Text

System	Process	Domain	Technique	Objective
Asium	semi -automated	Information extraction	linguistics and statistics	learn semantic knowledge from text
Text-To-Onto	semi -automated	Ontology management	linguistics and statistics	Ontology creation
TextStorm/Clouds	semi -automated	music and drawing	logic based and linguistics	build and refine domain ontology for musical pecies and drawings
Sndikate	fully automated	general ontology learning	linguistics based	build general domain ontology
OntoLearn	semi -automated	tourism	linguistics and statistics	develop interoperable infrastructure for tourism domain
CRCTOL	semi -automated	domain specific	linguistics and statistics	construct ontology from domain specific documents
Onto Gain	fully automated	general ontology learning	linguistics and statistics	build ontologies using unstructured text

2.10 Question-Answering Systems

A typical question-answering system aims to automatically answer user questions which are asked in a natural language syntax. Question-answering systems bifurcated, in term of applications domain, into two categories: open-domain [Hermjakob et al., 2000, Zheng, 2002b, Zheng, 2002a] and restricted-domain question-answering systems [Benamara, 2004,Katz et al., 2002]. Subject-oriented question-answering systems belong to the restricted-domain category. Generally, these systems are limited in terms of educational values, due to the poor quality of the returned answers [Feng et al., 2006]. On the other hand, general (open domain) question answering systems return good quality general

answers [Katz et al., 1993]. Domain specific question-answering systems typically return inaccurate answers due to the limitations of NLP approaches based on linguistic information [Gupta et al., 2008, Mollá and Vicedo, 2007]. Usually, restricted-domain questions fall within the hypothetical questions category which are more complicated than factoid and list question types. As a result, NLP techniques are not efficient for online learning environments, especially MOOCs, due to the large volumes of questions involved. Recently, question-answering systems for education and especially for online learning environments have emerged [Wen et al., 2012b, Mittal et al., 2005, Shatnawi et al., 2014, Wen et al., 2012a]. With the exception of our research work reported in this dissertation [Shatnawi et al., 2014], subject ontologies were not used in this research area.

2.11 Massive Open Online Courses (MOOCs)

MOOCs are a new phenomena in the higher education field. Despite attracting a great deal of attention in the last couple of years, there is very little research into the various aspects of MOOCs and their usage. In this section, MOOCs are described in detail and their features are outlined. Moreover, potential areas for research in MOOCs and the associated research challenges are discussed.

The development of MOOCs has its roots back to 2001-2002 when William and Flora Hewlett founded the Carnegie Mellon University Open Learning Initiative and the MIT Open Courseware project, which freely offered course materials from these institutions online under Creative Commons licenses [University, 2013]. The term MOOC was coined by David Cormier and Bryan Alexander at the University of Manitoba in 2008. In 2012 Edx which is a joint project between Harvard and MIT was established to offer open courses online; Udacity and Coursera also appeared in 2012. Currently, more institutions started offering MOOCs.

MOOCs have similarities to an ordinary course, such as a predefined timeline and a weekly breakdown of topics. However, MOOCs have no fees, no prerequisites other than Internet access, no predefined expectations for participation, and generally no accreditation, i.e. no credit or certificate offered for completion.

MOOCs have become a hot topic in higher education. E-learning and distance learning are well known concepts in the educational field. In addition, the use of technology such as radio and TV broadcasting, and the Internet, has been practised for some time. However, MOOCs are different in many aspects. Two of the most important characteristics are that MOOCs are free, i.e. institutions offer courses with no tuition fees, and that they are open, i.e. students can enrol with no prerequisite. The success of MOOCs is due to its adoption by prestigious institutions, offering opportunities to make education accessible and affordable, and to the availability of the Internet, tablets, and smart phones. As a result, we have the massiveness feature of MOOCs.

Higher education has many challenges. Among these challenges are: access, cost, and quality. MOOCs addressed and successfully resolved the access and cost challenges. However, the third challenge. which is quality, is the major controversial topic [Mazoue, 2013]. Some higher education researchers criticise the quality of MOOCs [Vardi, 2012]. Their view is that MOOCs lack a sophisticated learning architecture. In addition, they criticise the feedback and communication management in MOOCs. In current MOOC settings, instructors will not be able to interact with all students to answer their questions and comments. On the other hand, MOOCs support peer-to-peer interaction; however, this is not suitable for all types of courses [Mazoue, 2013].

Educational researchers who support the new phenomenon, see it as a solution for higher education challenges and a victory of democracy in education. They believe that the findings and the results of educational data mining, intelligent tutoring systems, and analytical learning researches will contribute to the

success of MOOCs and will enhance the communication and feedback management. Table 2.4 summarises the advantages and limitations of MOOCs based on pro/anti MOOCs perspectives [Mazoue, 2013, Vardi, 2012, Kaczmarczyk, 2013, Hyman, 2012, Cooper and Sahami, 2013].

TABLE 2.4: Advantages and limitation of MOOCs

Advantages	Limitations
More effective than a professor monologuing to a large class.	Inability of educators to assess student learning
It offers quizzes for retrieval practice which is an established method to improve learning	No accreditation
Open opportunities for millions of people who cannot access universities	Validation and plagiarism
Provides global access to education and can be scheduled to work with family and personal commitments	Lack of in-depth evaluation models to evaluate projects and assignments
Can be used as support materials for face to face courses	Lack of effective communication and feedback

According to Table 2.4, MOOCs open opportunities for millions of people who cannot access universities. Nevertheless, they may face challenges satisfying the cognitive needs of the massive number of registrants in these courses. The research presented in this dissertation aims to support MOOCs to overcome the lack of content related answers (feedback). Thus, our goal to offer automatic content-related answers for MOOCs registrants. In this research we target questions which belong to the four lower levels of Blooms's taxonomy ,namely, "Knowledge", "Comprehension", "Application", and "Analysis". [Bloom, 1956]

Chapter 3

State of the Art in Text Mining and Knowledge Engineering for Education

3.1 Introduction

It is worth emphasising that the work reported in this thesis falls in several areas which deemed to be a multidisciplinary research. As a result, it is important to explicitly define the boundaries of the literature review. This chapter reviews the state of the art research studies and methods related to our research which are mainly related to the ontology learning, short text classification, and question answering system. We defined the main concepts and terminologies of these different areas in Chapter 2. However, in this chapter, we provide thorough review of the state of the art methods directly related to the research conducted and reported in this dissertation.

3.2 Ontology in Education

Semantic web technologies can bring many advantages to the technology enhanced learning and can have profound effects on teaching and learning processes in both traditional and e-learning settings. Ontologies form the main component of the semantic web technology. However, the complexity of developing ontologies is one of the main limitations for adopting ontology-based applications in the technology enhanced learning. Since developing an ontology is notoriously costly and time-consuming task. Ontology editors, authoring tools, and maintenance tools are useful for creating and maintaining domain ontologies. However, these tools are not aimed to be used by novice and non IT users, such as educators. Even educators with good IT skills are not satisfied by these tool [Hatala et al., 2012]. Ontology development tools in the foremost support software engineers and don't hide the structural aspects of ontologies. Thus, Researchers in the technology enhanced learning start developing tools for the educational domain to automatically or semi-automatically extract and build ontologies from text and to hide the structural aspects of ontologies [Dicheva et al., 2005, Zouaq et al., 2007]. There are also a number of tools that leverage course content resources to automatically or semi-automatically build subject ontologies [Dicheva et al., 2005, Dicheva and Dichev, 2006]. These tools generally utilise the learning objects formatting characteristics such as chapter titles, headings, and sub-headings formats. Alternatively, some tools utilise table of contents and index structures. However, many learning objects don't include these characteristics. As a result, these tools fail to generate subject ontologies. Here is the gap that initiate this research, how to create a subject ontology from learning objects in plain text format?

Ontologies are not final products. Instead, they are components that support other services. For educational domain services, these ontologies can be (in fact it should be) connected to other domain ontologies to enhance the quality

of these services. Ontologies have been used in the educational field to represent course content [Crowley and Medvedeva, 2006, Zouaq et al., 2007, Boyce and Pahl, 2007, Chi, 2009, Zouaq et al., 2007]. It scaffolds students learning due to its role in instructional design and curriculum content sequencing [Coll et al., 2014, Chi, 2009]. Also, ontologies have been used in intelligent tutoring systems [Crowley and Medvedeva, 2006], student assessments [Litherland et al., 2013], ontology based user models [Razmerita et al., 2003], and feedback [Muoz-Merino et al., 2011, Shatnawi et al., 2014, Boyce and Pahl, 2007]. An ontology-based feedback framework to support students in programming tasks was introduced by [Muoz-Merino et al., 2011]. Existing ontology developing tools are categorised into three categories, which are: hand-crafting ontologies from scratch, semi-automatic ontology building, and search and retrieval of ontology from online resources [Hatala et al., 2012].

3.3 Question Answering System in the Educational Domain

Question answering systems for educational purposes fall in the specific domain question answering systems category. A part of the "learning companion" agent task [Goodman et al., 1997] is to answer student questions in technology-enhanced learning systems. In that research, question answering was not the final goal, it was just one of many tasks to support students learning by offering solicited feedback. The first large-scale educational question answering systems appeared in 2000 in a joint Special Interest Group for Linguistic Data and Corpus-based Approaches to NLP (SIGDAT) conference [Ng et al., 2000]. The goal of these systems is to answer comprehension questions about a specific reading passage. Many research studies validated their question answering systems using questions according to well defined standards [Hirschman and

Gaizauskas, 2001]. This type of questions can be classified as template questions according to [Carbonell et al., 2000] classification. In our research, we chose to validate our proposed question answering system using end of chapter questions; these question set by educators according to well defined standards. Although student questions in MOOCs discussion forums usually are not well formed and contains typos and some Internet jargons. However, we still can generate a well formed questions from the ill-formed questions in the question analysis phase given that we deal with a specific subject knowledge. The increasing advances of Internet technologies and the rapid development of matured e-learning technologies and services made the instructional benefits of computer supported collaborative learning apparent. Question answering systems are among these services. A question answering system to support collaborative learning was proposed by [Arai and Handayani, 2012], students can ask questions for their peers and also students can up-vote/down-vote an answer. However, this system doesn't offer an automatic answering service. As a result, this type of question answering systems is not appropriate for MOOCs due to the massiveness feature of MOOCs. Recently, some question answering systems for online learning and network education appeared [Zhang and Liu, 2009, Zhen and Zheng-wan, 2013]. However, these systems rely on manually created ontologies. Nevertheless, our question answering system relies on automatically generated subject ontology. The advantage of using ontologies for subject knowledge representation is that ontologies make the subject knowledge explicit which allows question answering system to use semantic reasoning to retrieve knowledge granularities and offers answers that meet well defined standards by educators. Question answering systems have many advantages for the technology enhanced learning and MOOCs. It mitigates effects of the information overload problem and it helps instructors who are usually overwhelmed by students questions and emails.

3.4 Short Text Classification

A comparative study for the effect of using different feature selection methods when applied to text classification was conducted by [Yang and Pedersen, 1997]. They found that the document frequency (DF) threshold is a reliable approach and that it had the lowest computational cost. Thus, DF can be used instead of information gain (IG) or χ^2 methods. Moreover, DF achieves consistent results for non-English texts [Xu et al., 2008b].

The advent of web 2.0 exacerbated the challenges in the text classification field. As a result, researchers proposed different feature selection methods to reduce the computational cost and to improve the effectiveness of existing text classifiers. [Mahajan and Sharmistha, 2015] proposed a Wavelet Packet Transform based feature selection; they use "HowNet" to expand the semantic feature space for keywords and phrases in short texts. On the other hand, unsupervised text classification was proposed by [Yin et al., 2015] and [Ezen-Can and Boyer, 2013].

A recent work proposed a methodology to analyse MOOCs discussion forums. They identified the purposes of the MOOCs registrants when they use MOOCs forums and the categories of their posts. In that research, they manually analysed the discussion forum data from the inaugural edX MOOC [Stump et al., 2013]. Although manual analysis of forums data is not appropriate for large scale data, their work is an important foundation for further automatic analysis. Following research studies focused on automatically understanding MOOCs discussion forum data from different perspectives. One study aimed to investigate the extent to which learners ask content-related questions and the extent to which facilitators answer these questions in MOOC discussion forums [Cui and Wise, 2015]. They found that students are not getting enough content-related feedback and they proposed a linguistic approach to identify content-related questions. An automatic forum discussion data analysis for improving student

retention and predicting student survival using a seeded topic model is explored by [Ezen-Can and Boyer, 2013,Ramesh et al., 2014]. They subsumed the categories identified by [Stump et al., 2013] into four broad categories and provide word seeds for each category to automatically capture them with the LDA topic modelling technique.

3.5 Summary and Perspectives

It is obvious that ontologies can support technology enhanced learning and improve the services of learning management systems. However, developing these ontologies remains a hurdle to adopt semantic web technologies in these systems. MOOCs can make use of ontologies to enhance the services they offer to their registrants. A possible service is the automatic question answering tool that mitigates the information overloading for MOOCs registrants. On the other hand, it is a relief for course facilitators who usually are overwhelmed by learners questions and emails. Typically, MOOCs registrants use discussion forums to ask their questions. However, these forums are used for different purposes and so it is important to filter (classify) those posts that contain questions and specifically content-related questions. Then, the question answering module processes these questions and send proper answers for MOOCs registrants. These are the scope of our research and we aim to answer the research questions through the proposed framework. The following chapters introduce the proposed framework, research methodology, and the system modules.

Chapter 4

The Proposed MOOC-Feedback Management System

4.1 Introduction

In this chapter, we introduce the research methodology to offer automatic timely content-related feedback for learners in MOOCs settings. We aim to build a comprehensive understanding for the proposed framework which makes the following chapters clear and highlights the importance of each module in the proposed framework. Also, we explain how these modules integrate to achieve the objectives of the research.

4.2 The Research Methodology

We used the model methodology to study and understand our research. We designed a model to run our experiments and to test the research hypotheses. We set a plan to evolve the proposed framework from small pieces to modules and

finally to integrate these modules together. First, we describe our methodology. Then, we introduce the proposed model.

4.2.1 Data Collection

In order to run our experiments we collected overlapping educational resources for the underlying subject. These resources include textbooks, slide notes, transcripts, and Blogs. We have two criteria for selecting these resources. First, it should comprehensively cover the subject. Second, it can be converted to plain textual format. These resources are consumed by the ontology learning module. Also, we collected discussion forums data for a MOOC. We collected posts that don't contain technical contents. We excluded any personal information to adhere to the data usage guidelines set by the offering institute.

4.2.2 Experimental Setup Design

In this section we introduce hardware and Software specifications. We used R statistical tools environment to run our experiments. We used the following R packages: tm, openNLP, Stanford coreNLP, Rtools packages for R. For some components we used Java and C++ programming languages to customise or build some of the proposed components. Details about the specific techniques and tools we used in our experiments are described in the following chapters. Our experiments answer the following questions:

- How can we leverage data mining techniques to build a subject course ontology?
- How can we improve the accuracy of short text classifiers using subject course ontologies?

- How can an ontology support question-answering system in MOOCs settings?

We validated every experiment using different measures including precision and accuracy measures. Also, we used comparative/benchmark validation approaches to validate the quality of the resulting subject ontology and the effect of the ontology-based feature indexing. We provide all details that allow other researchers to reproduce our experiments using different subjects resources and MOOCs discussion forums.

4.2.3 Reporting Experimental Results

We represent all results using tables and figures. We give a compelling explanation of these results. We show how different parameters affect these results. Each module has a discussion section to analyse the results and we offer self criticism of these results.

4.3 The Proposed Model

The proposed framework consists of three main modules which are the subject ontology learning module, short text classification module, and question answering module. The following is a brief description for these modules. Figure 4.1 depicts these modules and their interactions.

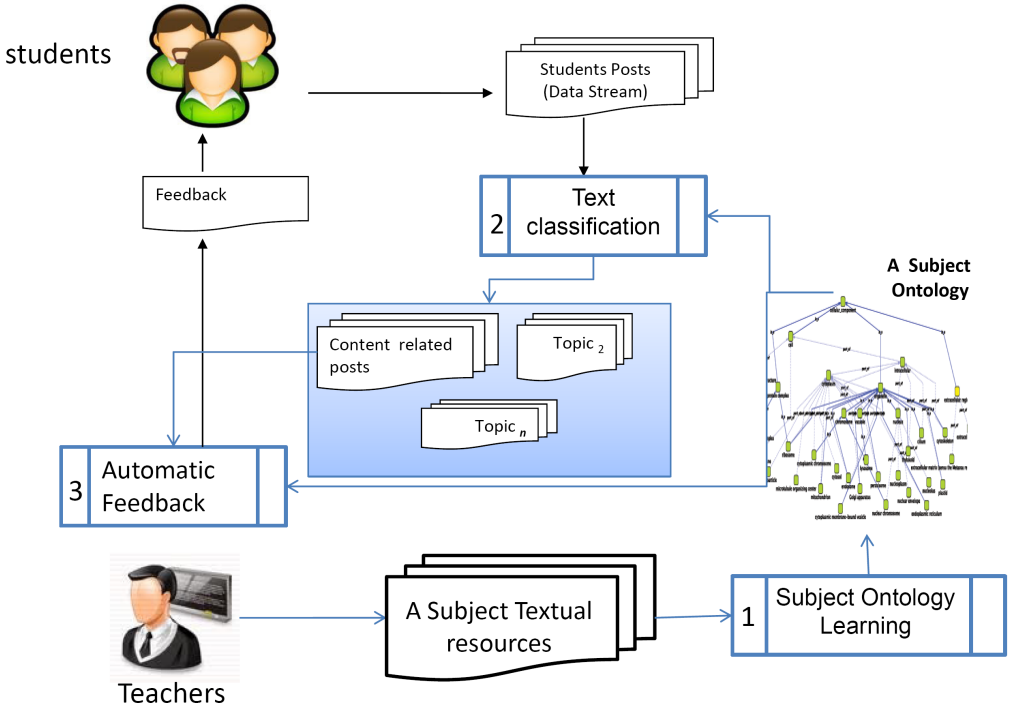


FIGURE 4.1: The Ontology based feedback framework

4.3.1 The Subject Ontology Learning Module

This module is the core module in the proposed framework. It supports other modules in the proposed framework. Building and maintaining ontologies is expensive and time-consuming task. To address this problem, semi-automatic or automatic approaches to building ontologies have emerged, which are referred to as ontology learning. These approaches focused on extracting concepts and relations from structured documents such as web page structures and book outlines and indexes, by using Natural Language Processing techniques. Unlike previous research, to identify relevant concepts for ontology learning, we used a regular expression parser approach widely adopted in compiler construction, i.e., deterministic finite automata (DFA). Our research is done in the context of

Massive Open Online Courses (MOOCs) and we used several overlapping heterogeneous learning resources for building the ontology. Thus, unlike previous research, the relevant concepts are extracted from unstructured documents. To build the concept hierarchy, we used a frequent pattern mining approach and employed a heuristic function to ensure that sibling concepts are at the same level in the hierarchy. As this process does not require specific lexical or syntactic information, it can be applied to any subject. To validate the approach, we employed the ontology in a question-answering system which analyses students' content-related questions and generates answers for them. We used a textbook end of chapter questions/answers to validate the question-answering system. Subject experts were asked to rate the quality of the system's answers on a subset of questions, and their ratings were used to identify the most appropriate semantic text similarity metric to use as a validation metric for the quality of the answers. Seven metrics were used and the Latent Semantic Analysis (LSA) was identified as the closest to the experts' ratings. We compared the use of our ontology vs. the use of *Text2Onto*(the state of the art tool). This module automatically builds a subject ontology that represent the subject knowledge. It consumes educational resources for the subject to generate the subject ontology. Chapter 5 describes this module which answers the first research question. In this module we have several novel contributions to the ontology learning field. First, we used the FP-tree algorithm in a new paradigm to construct the concept-hierarchy of subject ontology. We customised the FP-tree structure to fulfil the concept-hierarchy requirements. Second, we improved the quality of the resulting concept-hierarchy using a heuristic function based on association rule mining. Third, we represent the subject terms and phrases in DFAs and we constructed an automatic natural language DFA builder.

4.3.2 Short Text Classification Module

MOOCs registrants use discussion forums for different purposes and play different roles. Filtering content-related posts is an important action to identify registrants cognitive needs and to offer proper feedback for these posts. This module relies on the subject ontology to classify discussion forums data. We tested different stat of the art classifiers and we used different feature indexing approaches. We managed to enhance the adopted classifiers' accuracy. Chapter 6 envisages this module. We proposed two new feature indexing approaches for classifying short text documents. These indexing approaches improves the accuracy of the tested classifiers. These approaches are our contribution to the data mining field. We selected a number of text classifiers to test the effect of the feature indexing on these classifiers. We selected the top- performing, state of the art, classifiers. We namely select the following classifiers: Support Vector Machine (SVM), Neural Networks (NNet), Decision Trees (Tree), Random Forests (RF), Bootstrap Aggregation (BAGGING), and Supervised Latent Dirichlet Allocation (SLDA).

4.3.3 Feedback and Question Answering Module

This module offers automatic feedback (answers) for learner questions. It relies on most of the resulting components in the aforementioned modules. It consults the resulting subject ontology and the DFA component to parse student questions. In this step, the module identifies the topics and properties in the question. Then, it translates these constitutes into ontology triples. Finaly, it queries the subject ontology to retrieve answer parts. Finally, it aggregates

these parts and it displays answers to learners. Chapter 7 discusses this module in details. This module answers the third research question. Using subject ontology as a knowledge base to answer questions for educational purposes in MOOCs settings is our novel contribution to the technology enhanced learning field.

4.4 Illustration Scenario

In order to make the proposed framework clear, we introduce an illustration scenario. In this scenario, we demonstrate the processes of each component of the proposed framework. We build this scenario for a typical "Introduction to Database Management System" course. As a result, in each component we will show what the input, processing, and the output of the component are.

4.4.1 The subject Ontology Learning Module

First, we collected learning objects for a typical "Introduction to Database Management System" course. These resources include textbooks, slide notes, Wikis, and Blogs. Thus, we used the following books "Database Systems: Applicational Approach to Design, Implementation, and Management; 4th Edition","Database Management Systems; 2nd Edition." and "Fundamentals of Database Systems; 6th Edition". Also, we collected some slide notes for the course, and additional materials from Wikipedia[1]. These resources have different formats including PDF files, Powerpoint Slides, and HTML documents. We used a text convertor tool to represent all these resources into a plain text format. The conversion process may result in minor conversion errors. However, these conversion errors don't cause any significant effects on the ontology learning process. Next, we

[1]https://en.wikipedia.org/wiki/Database

found the frequent terms and phrases in the collected corpus. This step generated a large list of terms and phrases. Some of these terms and phrases are irrelevant to the "Database" subject. So, we used the COCA [2] [Davies, 008] corpus to augment these terms and phrases. We used the relative frequency method where a term or a phrase is removed from the list if its relative frequency in the corpus less than its relative frequency in the COCA. As a result, we have a list of augmented terms and phrases. Since it is not convenient to display the whole list of terms and phrases, we show an example of a subset of these terms and phrases in Table 4.1. The aforementioned steps are common in most ontology learning systems. Nevertheless, the following steps are novel. The obtained terms and phrases form the seed for the Deterministic Finite Automata DFA module. This module takes these terms and phrases as input. Then, it expands them using Wordnet package to get all the possible synonyms and generate a state table that represents all these terms and phrases. To illustrate this step assume that we have the following terms in the list which are: "database system" and "table", then the DFA module will generate all the possible synonyms such as "database management system", "database administration", "relation","file", etc. Next, for each phrase or extended phrase, it generates a DFA to represent that phrase. Typically, DFAs are used in compilers to parse programs and identify syntax errors in these programs. However, we used this tool in a new paradigm with natural language phrases instead of regular expressions. Collectively, these DFAs form unified "DFAs table" or what is called a "state table". Figure 4.2 depicts these steps. Now, we have the concepts for the "database"subject. Next, we should create the concept hierarchy for the "database" ontology. In order to achieve that we used data mining techniques. Specifically, we used frequent pattern mining techniques. We consider each paragraph in the textbooks a transaction. And each concept in a paragraph represents an item. As a result, we constructed the basket of transactions ("transaction database"). Then, we applied the FP-Tree algorithm

[2]Corpus of Contemporary American English

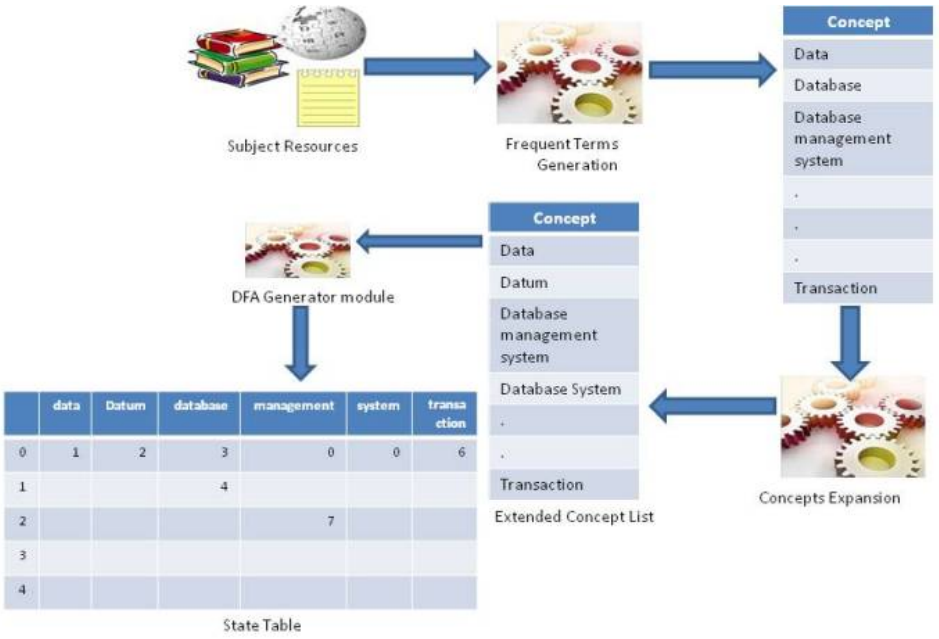


FIGURE 4.2: Example of extracting subject terms and concepts and building its DFAs table

on this transaction database. This approach is novel in the ontology learning field. However, the resulting FP-tree allows items (the subject concepts in our case) to appear many times. However, the concept must appear only once in the ontology concept hierarchy. As a result, we proposed another novel algorithm to fulfil this requirement by merging multiple items into one node. Figure 4.3 shows the aforementioned steps. Another problem appeared is that the hierarchy is not accurate due to the merging mechanism. Thus, we proposed a heuristic function based on the association rule mining. As a result we constructed a concept hierarchy as the one that is envisaged in Figure 4.4. After that, we attached a set of predefined properties to these concepts. Finally, we converted these concepts, properties, and the relationships among them into a OWL syntax. The resulting ontology supports the other components of the proposed framework.

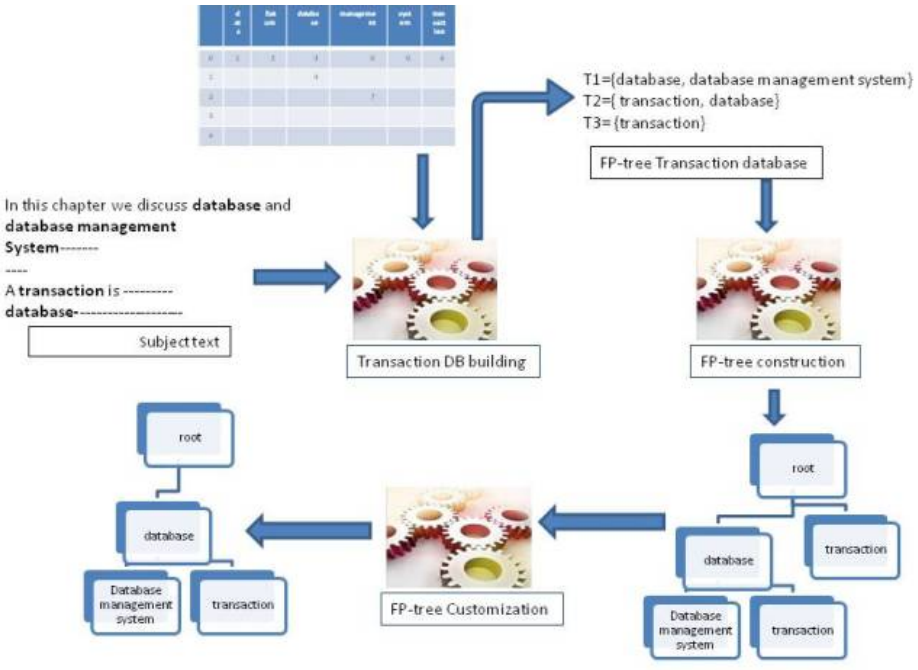


FIGURE 4.3: Example for building subject hierarchy form subject resources

4.4.2 Text Classification Module

As we aim to offer automatic answers for content-related questions in MOOCs forums, we should filter these questions out of other questions and comments in MOOCs forums. We proposed a novel indexing approaches based on the resulting subject ontology for classifying MOOCs discussion forums. For example questions like "Why the instructor wear a red shirt in all videos?", " Can you please introduce yourself?", or "How can I get the accomplishment certificate for this course?" will not be sent to the question answering system. On the other hand, questions like "What are the differences between a database system and a file system?", "What are the components of a database system?","How to create a table using SQL syntax?" will be passed to the question answering system.

Concept
data
database
database management system
select
sql
relationships
transaction
query
concurrency control
tuple

TABLE 4.1: A Subset of the "Database" subject frequent terms and phrases

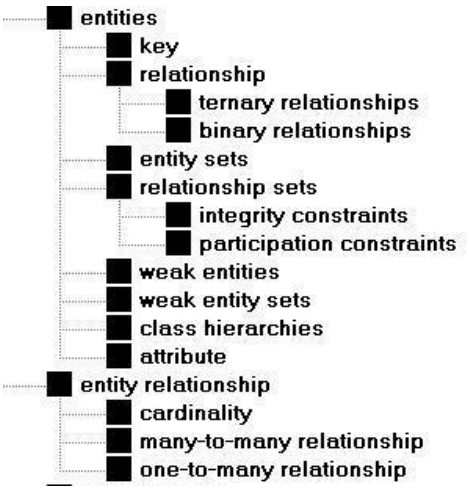


FIGURE 4.4: A subset of the "Database" Subject Concept Hierarchy

4.4.3 Question Answering Module

Finally, the question-answering module processes questions and then returns answers for the content-related questions. For example, if a student asks a question like "what are the components of DBMS?" Then the module will identify all concepts and properties appearing in that question by using the DFA state table and label these questions using the subject concept hierarchy. The

aforementioned question has "DBMS" concept and "components" property. It forms a query triple and uses the semantic reasoning approach to query the "database" ontology which extracts the answer stored in the ontology.

4.5 Summary and A Look Ahead

The proposed framework consists of three main modules. It represents a subject knowledge as ontology form. The other modules leverage the ontology representation to filter posts in MOOCs discussion forums. Then, it analysis these posts to generate proper feedback to MOOCs registrants. Some components are common like the DFA component which offers services to all modules. These modules integrate together to form the question answering system for subject content-related questions in MOOCs settings. The following chapters present these modules in details starting from the data collection phase till the final results. Every module has its own validation approach to ensure the correctness and applicability of the proposed framework.

Chapter 5

Automatic Subject Ontology Learning

5.1 Introduction and Objectives

Recent research in learning technologies took up existing semantic web knowledge and applied it to improve learning environments. This research includes educational data mining based on semantic web [Nayak et al., 2009], integrating educational resources with service-oriented architecture and web services using semantic web [Li and Wang, 2013], and semantic web applications for education [Kasimati and Zamani, 2011].

Ontologies form the main knowledge structure of semantic web. There is, however, a consensus among researchers that building and maintaining ontologies are expensive and time consuming tasks. In the learning technologies area most researchers either manually build a domain-specific ontology or assume the existence of such an ontology [Wen et al., 2012b, Li and Wang, 2013, Xu et al., 2008a].

Ontologies have been used in the field of learning technology for various purposes such as instructional design [Isotani et al., 2013], adaptive intelligent educational systems [Henze et al., 2004], tutorial dialog systems [Fiedler and Tsovaltzi, 2003], assessment [Kazi et al., 2010], feedback [Shatnawi et al., 2014] and question-answering systems [Vargas-Vera and Motta, 2004]. A comprehensive review of ontology use in e-learning systems can be found in [Al-Yahya et al., 2015] . Moreover, ontologies have great potential for supporting learning in the context of MOOCs (Massive Open Online Courses) by explicitly representing subject knowledge, which can then be used to offer personalised solutions. In the educational filed, in terms of technical solutions to facilitate ontology building, authoring tools for ontology creation dominate the research field (e.g. [Dicheva and Dichev, 2006, Yang et al., 2004, Aroyo and Dicheva, 2004], while semi-automatic [Zouaq et al., 2007] and automatic [Henze et al., 2004] approaches are less researched. In the wider ontology development field, there are tools for semi-automatic (e.g. [Kamel et al., 2013]) and automatic (e.g. [Cimiano and Völker, 2005]) ontology building; however, these tools were designed for IT experts, not educators [Hatala et al., 2012].

In this chapter we proposed an approach to automatically build a general subject ontology by leveraging data mining techniques. Unlike previous research, both in the educational domain and the wider ontology building area, we use overlapping educational resources. Moreover, while most of the previous research used linguistic approaches, we proposed a frequent mining approach that does not require linguistic information, which makes the proposed approach domain-independent.

The resulting ontology serves as an input to a question-answering system. To the best of our knowledge, there is no question-answering system for education that uses automatically generated ontologies as a knowledge base to answer questions. We hired domain experts to validated the proposed question-answering system. We used the convenience sampling approach in the validation

process [Gravetter and Forzano, 2015]. Also, we used different semantic similarity metrics to validate the returned answers and identify a suitable metric for wider validation (without the need for information from experts). Our experiments show that the LSA-based text similarity metric is the most suitable metric for validating the question-answering results.

We validated the subject ontology learning system through the results of the questions-answering system. We used a comparative validation approach by comparing the results when using our ontology with the results when using an ontology generated by Text2Onto [Cimiano and Völker, 2005], one of the most popular tools for ontology learning from textual resources.

5.2 Phase I: Ontology Building

In this section we present our proposed approach to automatically develop a subject ontology. We start by formally defining the general domain ontology, then we present our definition for a subject ontology and the purpose of developing the subject ontology. The proposed approach is described in detail for all the stages involved in the process.

An ontology is an explicit formal specification of a shared conceptualisation of a domain of interest [Studer and Staab, 2009]. An ontology defines the intentional part of the underlying domain, while the extensional parts of the domain (knowledge itself or instances) are called the ontology population. An ontology is formally defined in [Hotho et al., 2002]. Definition 3 formally defines an ontology.

Definition 3. A core ontology is a sign system $\Theta := (T, P, C^*, H, Root)$, where
T: a set of natural language terms of the Ontology
P: a set of properties
C^*: a function that connects terms t $\in T$ to a set $p \subset P$
H: a hierarchical organisation connecting all terms from T in a cyclic, transitive, directed relationships.
Root: is the top level node where all concepts in C^* are mapped to it.

Ontologies are mostly built upon a hierarchical backbone and bifurcate into two levels: upper ontologies that describe the most general entities and domain ontologies which describe a subject domain. Learning the upper ontologies from text is almost impossible in the foreseen future. However, the latter type can be extracted from textual resources. Although the formal ontology definition assumes the ontology to be described in intensional way as axioms and definitions in logic, In practice other types emerged which are prototype-based ontologies and terminological ontologies. Prototype-based ontologies are formed by collecting instances extensionally rather than describing the set of all possible instances in an intensional way. On the other hand, terminological ontologies describe concepts using labels or synonyms. Also, it is partially specified by subtype-supertype relations [Biemann, 2005]. Ontology learning for education presented different perspectives and had different purposes. So, to clarify our methodology and to build a common background for this research we will define our proposed ontology, identify its purpose, and introduce our motivation for developing a subject course ontology.

Definition: A subject course ontology is a formal representation of a subject contents that makes knowledge explicit.

Purpose: Learners consume learning contents to get knowledge. We aim to formally represent the contents of a particular subject to scaffold learning management systems in delivering course contents to learners. In particular, we aim to answer content-related questions.

Motivation: The massiveness property of MOOCs makes it difficult for the course facilitators to answer learners' questions in a timely manner. This increases the learner cognitive load and may increase the drop-out ratio. This motivated us to develop a general subject course ontology to serve as a knowledge base for an automatic answering system for the learners' content-related questions.

Type: The resulting ontology belongs to the terminological ontologies type. However, axioms and rules are added on the top of this ontology to allow semantic reasoning for the proposed question answering system.

Figure. 5.1 illustrates the proposed ontology development system. It shows the different phases to build a subject ontology and the packages we used or developed in every phase: (1) subject resources; (2) preprocess the data resources; (3) extract the subject terms; (4) construct the concepts hierarchy and apply our proposed heuristic function to enhance the quality of the concepts hierarchy; (5) export the concepts hierarchy into formal representation. In the following subsections we will describe these phases in details.

5.2.1 Data Collection and Preprocessing

In the educational domain, ontology learning research typically uses textbooks' table of contents, the structure of web pages or text formatting hierarchies to extract the underlying subject terms and to build the concept hierarchy for the

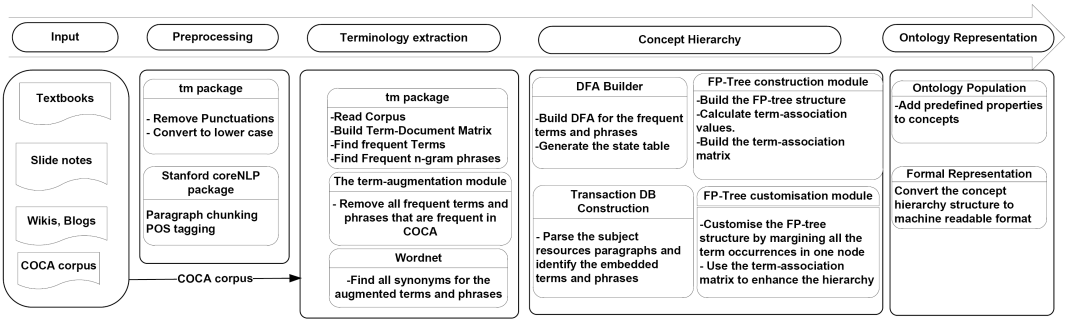


FIGURE 5.1: The Proposed Ontology Development System

underlying domain [Yan et al., 2009]. Many online and traditional educational resources, however, lack any given structure or text formatting hierarchy. As a result, the existing tools and techniques are not appropriate for these resources. We address this issue by building on the assumption that a subject ontology can be derived from heterogeneous overlapping learning objects (LOs) resources. These resources include textbooks, lecture notes, blogs, and other plain text subject resources. In this context, we do not need any knowledge about terms and the relationships among these terms, thus overcoming the limitation of lack of structure. This also allows a general approach to ontology building, from which ontologies for a variety of subjects can be built.

Generally, in the didactic domain, educators share a set of specific concepts and terms for a subjects' knowledge. As a result, when we collect overlapping resources for a subject, we can reveal that subjects' concepts. This assumption is suitable for the current MOOC settings where facilitators and learners have access to massive heterogeneous learning resources.

Educational documents provide definitions and explanations about concepts to be learnt. These concepts have typically low ambiguity and high specificity – for this reason learning objects are good candidates for building a subject course ontology. Textbooks share some characteristics when grouping concepts

together in learning units [Agrawal et al., 2016]. These characteristics support the proposed approach of learning ontologies. These characteristics are as following:

- Cohesion: Each learning unit consists of concepts that are closely related. For example concepts like "data", "information", and "knowledge" are colsely related and appear together in the "Introduction to Database" course for instance. While, "Normalisation","Concurrency Control", and "DML" are not tightly connected. As a result, related concepts appear together in learning units.
- Isolation: Concepts that belong to different learning units must be independent as much as possible.
- Unity: Some concepts, especially fundamental ones, may appear in different learning units.

We collected heterogeneous overlapping resources for the "Database Design and Management" subject. These resources are in different formats, some are PDF files, others are HTML pages, and the remaining resources are in MS Powerpoint slides. We converted all resources into PDF format, then we use a PDF text convertor tool to convert these resources into a plain text format. We stored all collected resources in plain text format and applied basic preprocessing to remove punctuations and other special characters from the text. However, stop words and numbers were not removed to allow meaningful part of speech (POS) tagging over these resources. It is worth mentioning that the conversion process may generate some minor errors. However, these errors do not cause significant effects to the ontology learning process.

5.2.2 Terms and Concepts Extraction

Terminology extraction is the process of discovering terms that are good candidates to represent the underlying domain in an ontology. It is the first and an important step in developing a domain ontology. Arguably, this is a matured phase and a plethora of techniques and measures exist in the literature. However, terms extraction for the education ontologies has not been examined to determine the best technique for developing a subject course ontology. In this research we used the terms frequencies (TF) and n-gram techniques to extract the key terms. We used the *"tm"* and *"RWeka"* packages for R to process the subject learning resources [Feinerer and Hornik, 2015a, Hornik et al., 2009]. Also, we used the COCA corpus (collection of documents) to filter the frequent terms [Davies, 008].

First, we built the document-term matrix (DTM) which is a two dimensional array data structure. A DTM describes the frequency of terms that occur in a corpus. Usually, rows correspond to words in the corpus and columns corresponds to documents in the corpus. The cell value describes the frequency of a word in a given document.

We used term frequency–inverse document frequency (TF-IDF), indicating the importance of a word in a document [Chowdhury, 2010], as the frequency weighting scheme as depicted in Equation 5.1 . All terms with frequencies above a given threshold θ are extracted as potential candidates for subject terms. The threshold value affects the number of the extracted terms and phrases. The larger the value is, the less the resulted terms and phrases are. We used a small threshold value which generates a large set of terms and phrases. However, all irrelevant terms and phrases are augmented in the following step. Since we uses overlapping subject resources, we expect to identify most of the subject key terms in this way. This approach is appropriate for educational documents, and our experimental results reported in Section 6 support this claim.

$$tf\text{-}idf_{t,d} \;=\; tf_{t,d} \times idf_t \;=\; tf_{t,d} \times log(\frac{N}{df_t}) \tag{5.1}$$

Where tf stands for term frequency; tf-idf stands for term frequency–inverse document frequency.

When we used the frequency measure to identify the ontology terms, we retrieved many irrelevant terms. In order to overcome this drawback we augmented the obtained terms based on the following approach: we assumed a term is a good candidate for a domain ontology if the term's TF-IDF value is greater than the term's TF-IDF in the corpus of the daily used terms. To achieve that, we used the Corpus of Contemporary American English (COCA) to get all frequent daily-used terms and phrases. The COCA corpus has more than 189,431 texts in the 450+ million word corpus (the last update to the corpus was in June 2012) [Davies, 008]. As a result, any frequent term in the underlying subject course corpus that is not one of the frequent terms in the COCA corpus is a candidate term for the subject course ontology.

We repeated the same approach with frequent n-gram terms, where n is the number of the words in a term ($2 \leq n \leq 5$). We set the maximum n-gram phrase to 5 words since our experiments showed that n-gram phrases that have 6 or more words are not frequent in the corpus even when we reduce the threshold θ to lower values. We extracted the n-gram phrases using *"RWeka"* package for R [Hornik et al., 2009]. Every frequent n-gram term in the underlying subject course corpus but not a frequent n-gram term in the COCA corpus is a candidate term for the subject course ontology.

In the context of the domain ontology learning, a concept is a semantic relationship among terms. Concepts differ from terms in that they are ontological entities that represent abstractions of human thoughts [Studer and Staab, 2009].

However, in a subject ontology a concept is a content unit or a learning objective a learner should learn or achieve. As a result, we can interchangeably use terms and concepts for a subject course domain ontology.

In the next step, we used "Jawbone Java API" through the Wordnet package for R to identify all synonyms for the candidate terms. Wordnet is a large English lexical database. It groups nouns, verbs, adjectives and adverbs into sets of cognitive synonyms called (synsets), where each synset expresses a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations [Wallace, 2007, Feinerer and Hornik, 2015b]. A possible disadvantage of this approach is that some concepts which are related to the subject course ontology may not appear in the extracted terms. However, we can allow educators or even learners to add any missing terms which is a task that does not require any technical expertise and can be achieved through a simple user interface. Algorithm 3 shows the pseudo-code for retrieving the subject ontology terms. The algorithm takes the subject course as input and returns a list of candidate terms and their synonyms.

5.2.3 Concepts Hierarchy Construction

In the ontology learning field, a number of research works used syntactic and semantic techniques to extract hierarchical relationships among the concepts of the underlying domain [Cimiano and Völker, 2005, Valencia-García et al., 2004]. However, recently there is a growing trend toward using machine learning techniques to construct hierarchical relationships among concepts. Researchers used Support Vector Machines [Wang et al., 2006, Li et al., 2005], Maximum Entropy Models [Zhang and Wang, 2012,Kambhatla, 2004] and Hidden Markov Models [Freitag and McCallum, 2000] to name a few.

In this research we used data mining techniques to extract the hierarchical relationships among concepts. We leveraged the characteristics of a subject

Algorithm 3 Extracting Subject Ontology terms

```
1: procedure FrequentTerms(corpus, terms)
2:     terms ← null
3:     Θ ← threshold
4:      COCA ← Corpus of Contemprory English
5:     DTM ← document terms matrix(corpus)
6:     terms ← freq terms(DTM,Tf-Idf,Θ)
7:     for (k = 2, k < 6, k++) do
8:         terms ← terms ⋃ freq(n − gram(DTM, k), Θ)
9:     end for
10:    for (t ∈ terms) do
11:        if freq (t) < COCA(t) then
12:            terms ← terms − t
13:        end if
14:    end for
15:    terms ← wordnetSynonyms(terms)
16: end procedure
```

course resources where intuitively related topics are grouped together or appear together in the contents learning resources. Specifically, we customised the frequent-pattern tree (FP-Tree) structure which was proposed by Han et al. (2000) and defined as in Definition 1 [Han et al., 2000].

Definition 4. Let C=$\{c_1, c_2, ..., c_m\}$ be a set of a course concepts.
DB=$\{T_1, T_2, ..., T_n\}$ a Transaction Database, where T_i ($i \in [1..n]$) is a transaction contains set of concepts$\in$C.
Let Support (S) be an occurrence frequency.
Let θ = minimum support threshold.
Then, P is a frequent pattern $\implies$ (P is a set of concepts $\in$ C) $\wedge$ S(P) $>\theta$.

In order to build an FP-Tree we need a transaction database (DB) and a minimum support threshold θ. We used textbooks as learning resources in this step. We considered every paragraph in textbooks as a transaction. All distinct concepts appear in a paragraph form the transaction items. In order to generate the transaction database for the subject course we split the corpus

into a set of paragraphs using *"openNLP"* package for R [Hornik, 2014]. The *"openNLP"* library is a machine learning based toolkit for processing of natural language texts written in Java. It supports the most common NLP tasks, such as tokenisation, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, and parsing. Also, we used the Stanford coreNLP library for co-reference resolution [Manning et al., 2014]. We parsed each paragraph in the corpus, and, as a result, we extracted the concepts appearing in that paragraph through the procedure explained in the following subsection. However, there are other options for building the transactions database such as using page level or sentence level segmentation instead of paragraph level segmentation. The former approach results in less transactions where each transaction contains more concepts. While the latter approach generates more transactions with less concepts in a transaction which in turn makes it difficult to build a reliable concept association matrix.

5.2.3.1 DFA Builder

In order to extract the concepts that appear in a paragraph we parse the paragraph word by word to discover all terms in a paragraph. To parse a paragraph, we built a deterministic finite automata for every term or concept extracted from the subject course textual resources. We considered every concept or any possible synonym a deterministic finite automata (DFA). DFA is formally defined in Definition 5. In our approach Σ is the set of all natural language words which are selected to represent a subject course ontology. We developed an automated DFA generator module that takes all concepts and their synonyms as input and generates a DFA for every concept and its synonyms.

Definition 5. A deterministic finite automaton (DFA) is a 5-tuple. (Q,Σ, δ, q0,F), where. Q is a finite set called the states, Σ is a finite set called the alphabet, $\delta : Q \times \Sigma \rightarrow Q$ is the transition function, $q0 \in Q$ is the start state, and $F \subset Q$ is the set of accept states.

TABLE 5.1: A Sample Mini State Table

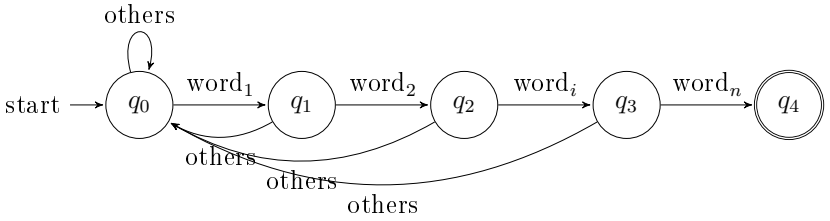
	input												
State	root	data	file	independence	item	model	types	warehouse	database	application	management	Others	Term ID
0	1	2	0	0	0	0	0	0	9	0	0	0	-1
1	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	1
2	θ	θ	3	4	5	6	7	8	θ	θ	θ	θ	2
3	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	3
4	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	4
5	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	5
6	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	6
7	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	7
8	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	8
9	θ	θ	θ	θ	θ	θ	θ	θ	θ	10	11	θ	9
10	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	10
11	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	θ	11

The module identifies all distinct concepts in the input list. Every word in a concept is a trigger to transfer the control to a specific state in the concept DFA. Fig. 5.2(a) shows an example of a DFA for a concept. Any concept consists of a number of words n where $1 \leq n \leq 5$. A DFA starts in the initial state q_0, each word causes a transition from a state to another state. If a word appears which does not belong to the concept words (others) then a transition to the initial state q_0 occurs.

Every DFA has a final state. When a DFA reaches a final state, it means that the DFA identified a concept. In Fig. 5.2(a) the state q_4 is the final state for that DFA. In an analog way, Fig. 5.2(b) shows another DFA for another concept. The state table generator module joins all DFAs and forms the state table. Fig. 5.2(c) shows an example of merging the DFAs of the two concepts c_1 and c_2. We assumed that both concepts start with the same first word ($word_1$). As a result, we merged the state q_0 and the state q_5. We repeated this step for the all obtained concepts and their synonyms. As a result, we generated the state table. An example of a state table is shown in Table 5.1.

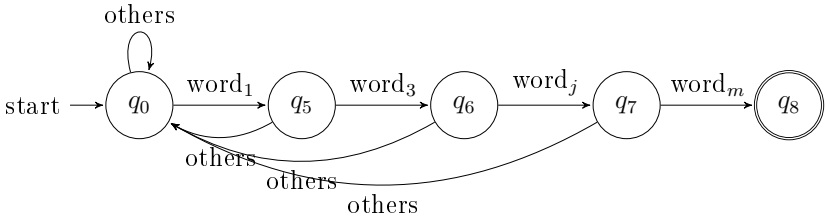
In the state table, columns correspond to words of the course concept list. On the other hand, rows correspond to the DFAs states. A cell has three possible states values which are a value of (0) represents an unexpected word which cause the parser to start again from state 0 (q_0), a positive number N, 0 <N <θ means a transition to state N, and a value of (θ) means a final state.If we

Let *c1* be a concept in a subject course ontology.
Let n be the number of words in *c1*.
Let "others"be any word $\notin$ *c1* words
Then the DFA that represents *c1* is:



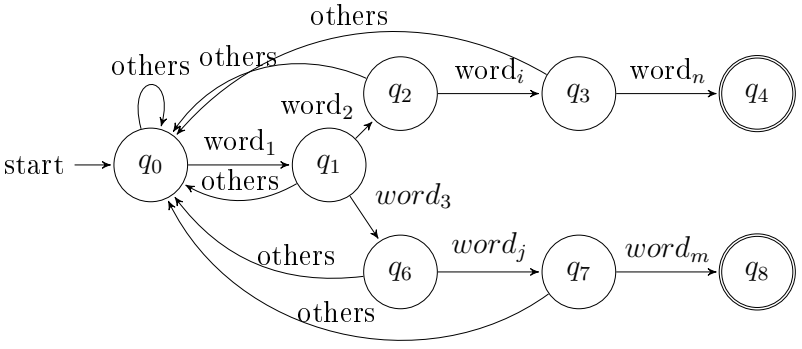
(A) Deterministic Finite Automata for concept c_1

Let *c2* be a concept in a subject course ontology.
Let m be the number of words in *c2*.
Let others any word $\notin$ *c2* words
Then the DFA that represents *c2* is:



(B) Deterministic Finite Automata for the Concept c_2

Suppose *c1* and *c2* share word1. i.e both concepts start by the same word
Then the DFA that represents *c2* and *c2* is:



(C) A Unified Deterministic Finite Automata for both concepts c_1 and c_2

FIGURE 5.2: Merging Deterministic Finite Automata for Concepts

TABLE 5.2: A Transaction Database

ID	Transaction [1]
1	C1, C2, C3
2	C2, C4, C5
3	C1, C2, C4
4	C1,C4
5	C1, C3

reach a final state then we identified a concept. The value in last column of a given final state (row) represents a term id.

In programming languages, compilers use this approach to parse program codes. However, we brought it in a new paradigm to parse natural language statements. Also, we automated the process of creating the state table to reduce any configuration complexity or human interaction with the system. This representation allows us to parse all words in a paragraph and to use phrases to index a paragraph. A paragraph may contain one or more concepts.

By using this approach portability is achieved since the state table for a subject course ontology is constructed automatically. As a result, the knowledge resources can be changed for a different subject and the ontology can be obtained (following the steps in the next subsections) with no extra configuration efforts as the state table is used regardless of the concepts it represents. Consequently, developing a new subject ontology requires only changing the learning contents resources.

5.2.3.2 Transactions database construction

The state table drives the parsing module to discovered all concepts in a paragraph. A paragraph generates a transaction. Each transaction consists of one

[1] C_i represents a subject concept

or more concepts. We add this transaction to the transactions database (DB). Algorithm 4 shows the pseudo code for extracting the transactions DB and Table 5.2 displays a subset of these transactions.

Algorithm 4 Generating Transactions DB

```
1: procedure TRANSACTIONSDB(corpus, Concepts, Transactions)
2:     paragraphs[] ← SplitCorpus(Corpus)
3:     Transactions ← null
4:     for (p=0, p<paragraphs.length(),p++) do
5:         Transactions[p] ← get all concepts(p)
6:     end for
7: end procedure
```

In an analog way, the state table is used to parse the user questions in the system-answering system – this is discussed further in the Section 5.3.

5.2.3.3 FP-Tree construction

The FP-Tree algorithm takes the transaction database as input to generate the FP-Tree structure shown in Fig. 5.3(b). A header table is constructed which contains all the items in the transactions DB with their corresponding frequency (count). It also contains a pointer to the first occurrence of an item (concept) in the tree. Thus, every node in the tree has a pointer to the next node occurrence in the tree. By applying the FP-Tree algorithm [Han et al., 2000] illustrated in Algorithm 5, the FP-Tree structure in Fig. 5.3(b) is obtained, where every node in the tree corresponds to a concept and its frequency count.

5.2.3.4 FP-Tree customisation

An item (concept) may appear many times in the original FP-Tree structure. However, in the ontology structure any concept should appear only once in

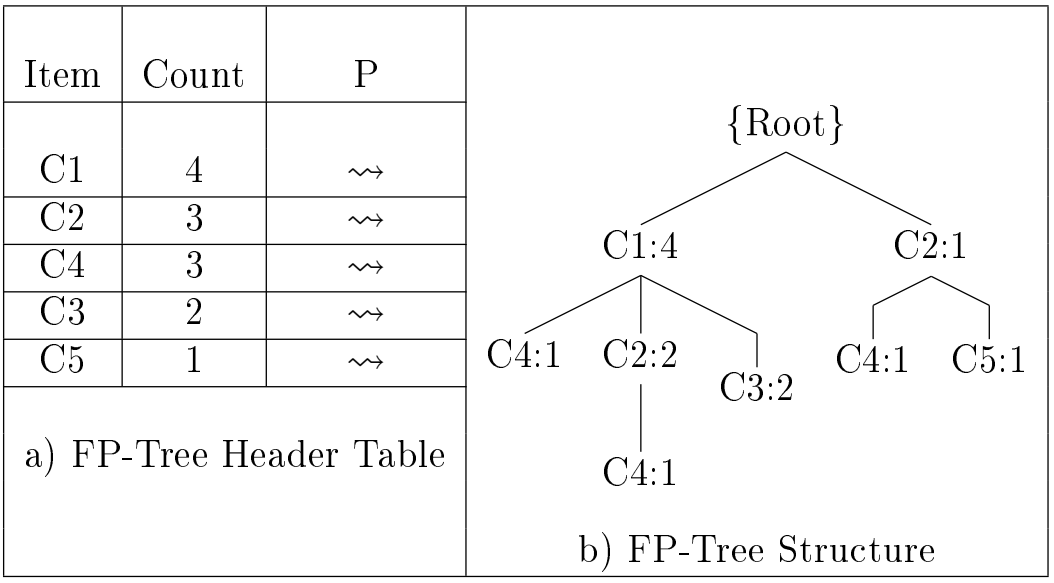


FIGURE 5.3: FP-Tree Construction

the concept hierarchy. As a result, multiple occurrences of a concept should be removed. To fulfil this ontology structure requirement, we customised the FP-Tree structure by merging multiple concepts into one instance.

The criterion used for merging concepts is their frequency. All concepts will be merged under the concept's instance that has the maximum frequency. A top down approach was followed in merging these concepts, by parsing the tree starting from the concept with the highest frequency (top level) down to the lowest frequencies (leaves). All descendant concepts are merged under the concept at the highest level in the structure.

This process may generate a hierarchy where sibling concepts appear in a parent-child hierarchy, i.e. concepts may be pushed down to the lower levels in the concepts hierarchy. To overcome this problem, we apply a heuristic function to determine if a concept should be moved to become a sibling of another concept. This decision is based on the term-association matrix, which is

TABLE 5.3: A Sample of A Term-Association Matrix

	conceptual	conceptual schema	data	data model	database	database application	dbms	entities	entity relationship	programs	queries	query language	rdbms	relational data	relations	relationship	schema
conceptual schema	1.13208																
data	2.64151	2.64151															
data model	1.50943	1.13208	6.03774														
database	2.26415	1.50943	10.9434	4.5283													
database application			4.15094	1.50943	4.5283												
dbms	1.50943	2.64151	18.4906	4.90566	12.4528	4.5283											
entities	1.13208			1.13208	1.13208												
entity relationship	1.13208		2.26415	2.26415	1.88679			3.77358									
programs			5.28302	1.13208	3.01887	1.50943	6.79245										
queries			5.28302	1.13208	3.39623	1.50943	4.15094			1.13208							
query language			3.01887		1.50943	1.13208	3.39623			1.50943	3.39623						
rdbms			2.64151	1.13208	1.13208	1.88679	2.26415					1.13208					
relational data			1.13208		2.26415	1.13208	1.88679		1.13208		1.50943						
relations	1.50943	2.64151	4.5283	1.13208	4.15094		3.77358				3.01887	2.64151		1.88679			
relationship	1.13208		1.13208	1.50943	1.50943			10.9434	3.77358								
schema	3.77358	2.64151	6.41509	3.01887	5.66038		4.15094	1.13208	2.26415	1.13208				1.88679	3.77358		
tuple		1.13208	2.64151	1.50943	3.77358		4.15094	2.26415	1.50943		1.13208	1.13208			4.15094	2.26415	3.39623

obtained by transforming the output of the FP-Tree algorithm in a symmetric matrix from, where the row/column are concepts and the values represent the associations between concepts. Table 5.3 shows a sample of the term-association matrix.

If the association value between the current node and its parent is lower than the association value between the current node and its grandparent, the current node is promoted one level up in the concept hierarchy. Consequently, the current node and its original parent become siblings in the hierarchy. An example of this process in given in Section 5.3.

Algorithm 6 shows how the FP-Tree structure is refined to solve multiple occurrences of concepts and the siblings problem. A top-down traversal is used for the first aspect, while a bottom-up traversal is used for the second one. We can generalise the proposed system for any subject course resources. As a result, we reduce the complexity of processing subject resources (learning objects). The entire process of generating a subject ontology is illustrated for a particular subject in Section 5.3. Incorrect concept hierarchy propagates to the question answering system. It labels questions incorrectly. As a result, the correctness of answers is affected. So, we can validate the quality of the resulting concept hierarchy using "precision" and "recall" measures of the question answering system.

Algorithm 5 FP-Tree Construction

```
1: procedure BuildFPTree(DB, θ)
2:     for (i=0, i <DB.length(),i++) do
3:         for (j=0,j <T_i.length(), j++) do
4:             Frequency(cj)++
5:         end for
6:         for (k=0,k <C.length(), k++) do
7:             if Frequency(c_k) >θ then
8:                 FrerquentTerms ← c_k
9:             end if
10:        end for
11:    end for
12:    L[] ← Sort (Frequent Concepts, DES)
13:    root ← new node(FP-Tree)
14:    for (i=0, i <DB.length(),i++) do
15:        Select frequent concepts ∈ t_i
16:        Sort (t_i) based on L
17:        current_node ← root
18:        for (t=0;t<t_i.length()) do
19:            if c_t ∈ current_nod.childern then
20:                current_node.child(c_t).count ++
21:                current_node ← current_node→child(c_t)
22:            else
23:                new current_node(c_t)
24:                current_node(c_t).count =1;
25:                current_node ← current_node→child(c_t)
26:            end if
27:        end for
28:    end for
       Description
       DB: The transaction Database
       T_i: A transaction in DB
       C: A set of all concepts (items).
       c_i: A concept in C.
       L: Header table contains all concepts sorted according to the concept frequency in descending (DES) order.
29: end procedure
```

* A detailed FP Tree algorithm as proposed by Han et al. 2000

Algorithm 6 Building The Concept Hierarchy

```
1: procedure ConceptsHierarchy(Transactions)
2:     FPTree ← Build FP-Tree(Transactions[])
3:     for c ∈ Concepts do
4:         SourceNode ← c
5:         for node ∈ Nodes(c) do
6:             Merge(SourceNode,c)
7:             for child ∈ child(c) do
8:                 Parent(child) ← SourceNode
9:             end for
10:        end for
11:    end for
12:    for node ∈ Nodes(c) do
13:        A ← Association(c, Parent(c)
14:        B ← Association(c, GrandParent(c)
15:        if A <B then
16:            Parent(c) ← GrandParent(c)
17:        end if
18:    end for
19: end procedure
```

5.3 Experimental Work and Results

In order to test our proposed system, we collected overlapping learning objects for the "Database Design and Management" course. These resources are combinations of book chapters[2][3][4], slide notes, blogs, and Wikis [5] . All resources are stored in plain text format.

5.3.1 Terms Extraction

The system found all frequent words in the corpus, as well as bigram, trigram, 4-gram, and 5-gram frequent phrases. All frequent terms that are not frequent in the COCA dictionary were selected to represent the course ontology as described in Algorithm 3.

The "Wordnet" library was used to retrieve all possible synonyms of the extracted concepts. We found that this step generated many irrelevant terms. A possible reason is that terms and concepts in a subject domain are used in more specific contexts than their general meaning. For example, the term "table" is used to describe the data structure for storing data in relational databases; however, synonyms like "bench", "worktop" or "counter" are not used in the context of the relational database subject to describe the same data structure. These extra synonyms did not significantly affect the quality of the domain ontology, but resulted in an increase of computation complexity of the subsequent steps. Table 5.4 shows a subset of the terms extracted after implementing this phase.

[2]Database Systems: Applicational Approach to Design, Implementation, and Management; 4th Edition

[3]Database Management Systems; 2nd Edition.

[4]Fundamentals of Database Systems;6th Edition

[5]https://en.wikipedia.org/wiki/Database

TABLE 5.4: Subset of The Database Design and Management terms

ID	Term
1	backup
2	buffer
3	calculus
4	client server
5	codd
6	commit
7	conceptual data
8	conceptual schema
9	concurrency
10	concurrency control
11	data
12	data entry
13	data file
14	data independence
15	data item
16	data model
17	data structures
18	data types
19	data warehouse
20	database

We used the list of frequent concepts and their synonyms as input to build the state table through the use of the deterministic finite automata (DFA) structure, as explained in Section 5.2.3.1.

To illustrate this step we refer to the concepts in Table 5.5. For simplicity we omitted the synonyms of these terms. We built a DFA for every concept and obtained the state table shown in Table 5.1, Section 5.2.3.1. This state table is used to parse the user questions. Example 1 illustrates the process of parsing a natural language statement using the state table.

Example 1. Parsing an NLP statement using the state table If we have the following statement "in database, data model is ..." then this statement is parsed and checked against the state table.

Input:"in database, a data model is ..."
Tokens: [in, database, a , data, model]
state_ table: Table 5.1, Section 5.2.3.1 – based on the concepts in Table 5.5.
State: is the current DFA state. Initially state=0. The first column in the state table holds the state values.
Steps :
- The first token is "in". We look for its value in the $state\_table[state = 0, \text{“}in\text{”}]$; as the word "in" is not a column in the *state_ table*, the value is taken from the column "others" (see also Fig. 5.2). Consequently, for the word "in", the value of $state\_table[state = 0, \text{“}others\text{”}]$ is 0, which means that this word is ignored and the parsing starts again from state 0 with the next token.

- The next token is "database", for which the value of $state\_table[state = 0, \text{“}database\text{”}]$ is 9, which means go to state 9. The next token is "data", and thus, we find $state\_table[9, \text{“}data\text{”}] = Acc$ indicating that a final state was reached. Reaching a final state denotes that a term was found, which can be identified from the Term ID (last column in Table 5.1); in this example the term ID is 9, which can be found in Table 5.5 to be "database";

-We continue till the end of the statement. The result of this step is that we identified all term IDs which are mentioned in the natural language statement.

Through the process mentioned above, another term with the ID 6 is identified, which corresponds to "data model" in Table 5.5. Thus, for the example above, two terms were identified.

We used this state table to parse the course learning resources to identify the subject concepts and to create the transactions DB for the FP-Tree module.

TABLE 5.5: Sample of The Extracted Concepts for The "Database Design and Management" Course

ID	Term
1	root
2	data
3	data file
4	data independence
5	data item
6	data model
7	data types
8	data warehouse
9	database
10	database application
11	database management

5.3.2 Concept Hierarchy

To create the transactions DB for the corpus was divided into paragraphs using the "openNLP" library and the co-references were resolved by using Stanford "coreNLP" library.

Each paragraph was parsed using the state table. As a result, each paragraph will add a transaction to the transactions DB. A transaction contains all term IDs which appeared in that paragraph. As a result, we obtain the transactions DB.

In the next step, the FP-Tree algorithm (see Algorithm 6) was used to build the FP-Tree structure. The algorithm gives as an output term-term association values, which have been stored in a term-association matrix – Table 5.3 shows an extract of this matrix.

We used the generated FP-Tree structure and the term-association matrix to enhance the quality of the concept hierarchy by merging co-occurrent concepts in the tree structure and by solving the siblings problem. As a result, the concept hierarchy is obtained – Fig. 5.4 shows part of the obtained concepts

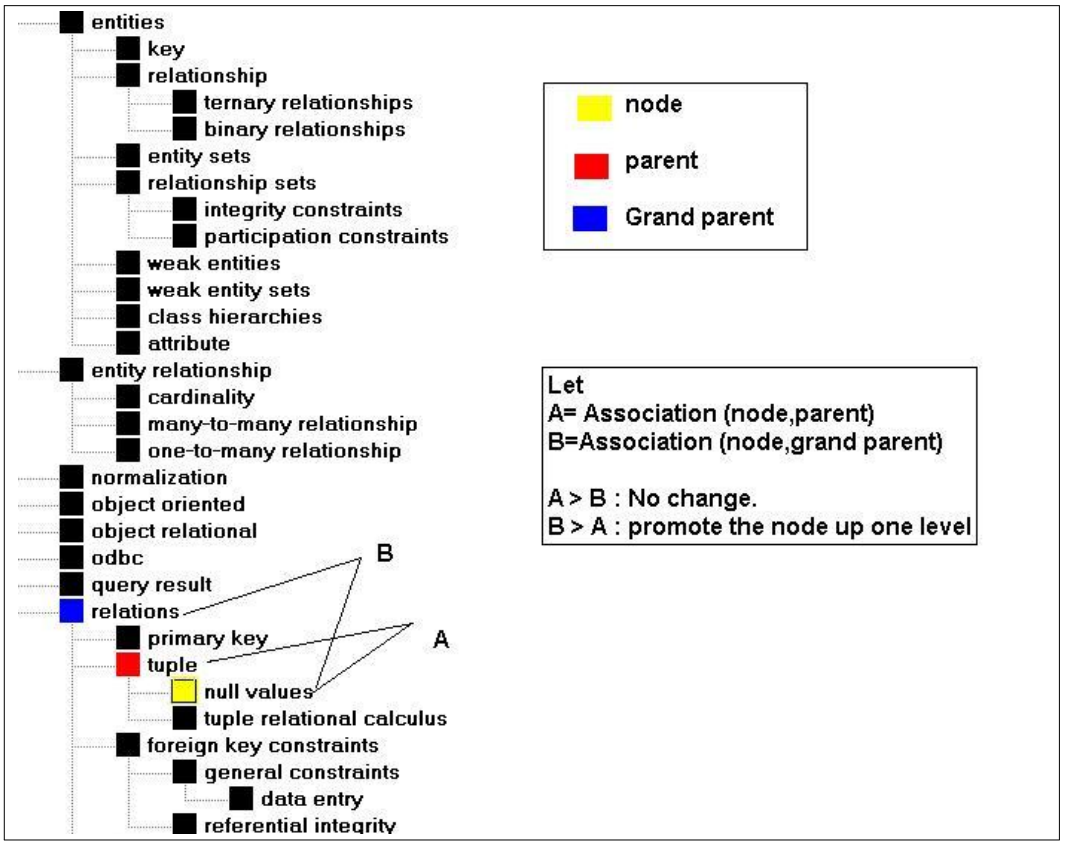


FIGURE 5.4: Sample of The Concept Hierarchy

hierarchy, as well as . the heuristic function used to sort the siblings problem. As aforementioned, the resulting ontology is of a terminological ontology type. Most relationships in this type are either *is–a* or *has–a* relationships. However, for the sake of the question answering system, we added other relationship types as proprieties which is a possible approach in ontology building. We discuss these properties in Chapter 7 . The final step was the formal representation. The OWL syntax was used to formally represent the subject ontology (concept, property, feedback) triplets as illustrated in Fig. 7.2.

TABLE 5.6: Experimental Result Summary

Questions	Count	Percentage
Answered	78	79.6 %
Not Answered	20	20.4 %
Total	98	

5.4 Validation

To validate the generated ontology we measured the impact of using this ontology on the question-answering system for answering content-related questions. The end of chapters questions for the "Database Design and Management" textbook [Connolly and Begg, 2001] were used to test the system. The contents of this textbook were intentionally left out when building the ontology.

The performance of the question-answering system using the proposed subject ontology was compared with the performance of the system when using an ontology produced with the Text2Onto tool.

To evaluate the answers given by the question-answering system, we compared them with the answers from the textbook for 98 questions. The system answered 78 questions out of 98 content-related questions. The system was not able to answer 20 questions because their subject terms were not represented in the subject ontology. As a result, all missing terms in the subject ontology will result in no answer for any question related to these terms. Table 5.6 shows the percentage of answered/not answered questions.

To identify the best metric for assessing the similarity of the answers, 5 subject experts (who taught "Database design and management" at university level in 5 different universities) were asked to evaluate the answers to 10 random questions on a scale from 1 (irrelevant/wrong) to 5 (relevant/accurate). Table 5.7 shows the summary of the expert evaluations.

TABLE 5.7: The Expert Evaluations Summary

Question	Mean	SDEV	LSA similarity
Q1	4.86	0.38	0.643
Q2	4.00	1.15	0.633
Q3	3.86	1.07	0.645
Q4	3.14	1.86	0.259
Q5	4.00	1.15	0.484
Q6	3.67	1.21	0.354
Q7	3.14	2.04	0.83
Q8	4.50	0.55	0.623
Q9	4.71	0.76	0.896
Q10	4.43	1.13	0.594

To identify the best metric for text similarity, we used the following 7 metrics: greedy comparison based on Wordnet introduced by [Lintean and Rus, 2012] to measure the semantic similarity between texts, Latent Semantic Analysis (LSA) using TASA corpus, optimal matching using LSA and TASA corpus, greedy paring using LSA and TASA corpus, greedy comparison using Latent Dirichlet Analysis (LDA) and TASA corpus, Corley and Mihalcea comparer (CM comparer) [Corley and Mihalcea, 2005] and bilingual Evaluation Understudy (BLEU) which is an automated method to evaluate machine translation from a language to another introduced by [Papineni et al., 2002] (which can be extended to find the similarity between texts). These were implemented using the "Semilar" toolkit [Rus et al., 2013].

Generally, greedy methods calculate the similarity score between TextA and TextB by pairing every word in TextA to all words in TextB. Then, a similarity metric is used to find word to word similarity. Finally, it greedily returns the maximum similarity score between TextA and TextB. The optimal comparer methods represent TextA and TextB as a weighted bipartite graph and find a matching from TextA to TextB which has the maximum weight [Rus and Lintean, 2012].

TABLE 5.8: Pearson Correlation between Similarity Measures and Experts evaluations

Similarity Method	Pearson Correlation
Greedy Comparer WNLin	-0.12
CM Comparer	-0.02
LSA	0.81
Optimum LSA/Tasa	0.12
Greedy LDA/Tasa	-0.11
Dependency WordNet Lesk/Tanim	0.41
BLEU Comparer	0.06

In order to determine the most appropriate measure for our system we used the aforementioned text similarity measures to calculate the similarity between an answer returned by our question-answering system and its answer key which is provided by the textbook authors.

We used the answers evaluated by experts to benchmark these different measures. We removed the extreme values (which have significant standard deviation), i.e. Q7 where the standard deviation is 2.04, and then calculated the Pearson correlation factor as defined in Equation 5.2.

$$r = \frac{\sum (x - \bar{X})(y - \bar{Y})}{\sqrt{\sum (x - \bar{X})^2 (y - \bar{Y})^2}} \tag{5.2}$$

where $\bar{X}$ and $\bar{Y}$ are the means of the data sets X and Y respectively.

Table 5.8 summarises the Pearson correlation factor between the text similarity measures and the experts evaluation. The best correlation score of 0.81 is achieved by the LSA based similarity metric. Therefore, the LSA based similarity metric was adopted in the validation step to calculate the similarity score between an answer generated by the question-answering system and its corresponding answer key provided by the textbook authors.

Next, we introduce LSA in more details since we adopt it to be the main similarity metric in validating the returned answers. We used "Semilar" system which is a text similarity tool based on Latent Semantic Analysis (LSA) [Rus et al., 2013].

LSA processes a matrix to produce three matrices. This matrix is usually a document-term matrix. The column indexes correspond to the documents in a corpus and the row indexes correspond to the terms in these documents $M_{i,j}, 0$ <i <d , 0 <j <t, d,t >0, d is the number of documents in the corpus, t number of different terms in the corpus. It uses the singular value decomposition (SVD) technique which is formally defined in Definition 6 to decompose M into three matrices T, S and D.

Definition 6. Let M be a matrix with $d \times t$ dimensions then M can be divided into
$M_{d\times t} = T_{t\times n} S_{n\times n} D_{n\times d}$
Such that T and D are orthonormal columns and S is diagonal. Then this is called singular value decomposition of M.

Usually S contains positive values sorted in descending order. SVD allows a simple strategy for an optimal approximation fit using smaller matrices. It uses the maximum k singular values in the matrix S and sets the remaining values in the S to zero. Accordingly, it selects the first k columns of the matrix T and the first k rows of the matrix D. Then, it represents the matrix M using the new augmented matrices as in the following formula: $M \approx M' = T'_{d\times k} S'_{k\times k} D'_{k\times t}$.

Applying a Singular Value Decomposition (SVD) on the term-document matrix results in an approximation of it using only the largest k singular values of the decomposition. This represents the LSA model, which is used to find the semantic similarity between words. It can be extended to find similarity between documents by aggregating the semantic similarity measures for all words in these documents. LSA is an effective tool in detecting word to word similarity

TABLE 5.9: LSA Based Similarity (Answer vs Answer Key)

Range	Count
0- 0.10	2
0.10 - 0.20	0
0.20 - 0.30	2
0.30 - 0.40	2
0.40 - 0.50	1
0.50 - 0.60	2
0.60 - 0.70	14
0.70 - 0.80	2
0.80 - 0.90	12
0.90 - 1.00	41

beyond the lexical word to word synonyms. LSA leverages the idea that the aggregate of all the word contexts in which a given word does/does not appear provides a set of mutual constraints that largely determines the similarity of meaning between words and sets of words.

We used the LSA text similarity tool ("Semilar") to compare the answer keys of the end of chapter questions to the 78 answers returned by the proposed question-answering system. Table 5.9 shows the similarity summary. We divided the table into 10 ranges; for every range, we count the number of answers that fall in that range. For the 10 questions evaluated by the experts, the last column in Table 5.7 shows the LSA based text similarity between the answer key and the automatic generated answer pairs.

There are 71 out of 78 answers (91%) with a value above 60% for the LSA metric. Moreover, the majority of the answers (53 answers representing 68%) have similarity values above 80%.

A possible reason for having answers which have a low similarity ratio is that these questions ask about multiple concepts and some of these terms were not selected among the subject ontology terms. As a result, the system will answer part of the question and ignore the remaining part of the question. In fact,

this occurred for questions 3 and 4 of the ones evaluated by the experts. Some concepts were not listed in the subject course ontology due to the following reasons:

- These concepts are not frequent concepts in the corpus used to build the ontology;
- These concepts are frequent in the corpus, however, they are also frequent in the COCA corpus; as a result, the proposed ontology system will remove these concepts from the ontology concept list;
- These concepts are synonyms that have not been generated by the "Wordnet" synonyms tool.

This drawback can be overcome by allowing course facilitators to add any missing concepts to the concept list. This task does not require any technical experience. Also, since we proposed an automated state table construction module, the following modules in the subject course ontology system do not require any modification.

Finally, we used the comparative validation approach [Zouaq and Nkambou, 2009] to validate the generated ontology. We ran the Text2Onto tool [Cimiano and Völker, 2005] on the same corpus to generate a subject ontology and used the generated ontology in the question-answering system to answer the 98 questions. Table 5.10 shows the accuracy (i.e. percentage of answered questions) of the question answering system using both Text2Onto and our proposed ontology.

Our approach outperforms the Text2Onto tool. We noticed that the Text2Onto tool generated a long list of irrelevant terms comparing with our proposed system, which affected the quality of the generated ontology. As a result, the question answering system performed poorly when using this ontology.

TABLE 5.10: QA Accuracy using TEXT2ONTO and the generated ontology

Ontology	Answered	Not Answered	Accuracy
Text2Onto	28	70	28.6%
Proposed Ontology	78	20	79.6%

5.5 Summary and Conclusions

In this research, we proposed a framework to automatically build a subject ontology from overlapping heterogeneous learning contents in plain text format. We represented the subject terms and concepts using the Deterministic Finite Automata (DFA) notation. We developed a module that takes the subject terms and concepts and generates a state table for these terms and concepts. This state table is used in the following modules to detect the subject concepts for the concept hierarchy construction and for parsing the questions in the question-answering system.

We used data mining-based techniques in novel approach to construct the concept hierarchy for the subject concepts. A heuristic function based on concept association mining drives the concept hierarchy construction module to enhance the quality of the concept hierarchy structure by resolving multiple occurrences within the hierarchy and by solving the siblings problem. The DFA representation and the concept-hierarchy construction modules make our approach applicable to different subjects.

The proposed ontology learning systems is suitable for e-learning environments, especially for MOOCs settings and educators with novice IT skills. We validated the resulting ontology using comparative validation approach by comparing it to the resulting ontology using the Text2Onto tool, a popular state of the art tool for learning ontologies from text. We used the resulting ontology in a question answering system. We validated the results using the subject course experts and using an LSA based text similarity metric. We used a set of content related questions from a "Database Management Systems" textbook to test the

question-answering system and evaluated the quality and the correctness of the returned answers. The results support our hypothesis, as the system was able to correctly answer 79.6% of the questions, which is significantly more than the 28.6% obtained when using Text2Onto. Text2Onto failed to capture many important concepts. Moreover it created a flat concept hierarchy where most of the concepts were organised directly under the root node as siblings. As we aforementioned, incorrect concept hierarchy leads to incorrect returned answers. These results proved that our proposed concept hierarchy approach effectively captures the correct subject concept hierarchy. However, there is a room for improvements since we achieved 79.6 % and this could be a future investigation. Another limitation of our approach occurs when the system fails to capture some concepts of the underlying subject. It propagates to the Q&A module where missing concepts are also not captured in students' questions. As a result, it will generate an incomplete answer (a partial answer) to that question. This explains the low similarity values in Table 5.7 for the questions 3 and 6. Also, the complexity of the questions may affect the quality of the generated answers. Questions at the higher levels of Bloom's taxonomy may not be answered correctly as occurred in question 7 in Table 5.7. However, we can overcome this limitation by initiating a dialogue with the learner to ask them to split their questions into multiple sentences.

There are many opportunities to use the proposed system in MOOCs. The resulting subject ontology can support pedagogical agents to support both collaborative and individualised learning, as well as the students' cognitive processes. Ontologies can be used to adapt learning units to a learner's profile. Subject ontologies support short text classification and clustering. It can be used to cluster MOOCs discussion forums and offer quantitative and qualitative analysis of these discussions for MOOCs facilitators . On the other hand, the Q&A system can be extended to analyse students cognitive needs and give feedback for course facilitators about students learning. Ontologies can also be used in learners' assessments .

Chapter 6

Short Text Classification Module

This module uses the resulting subject ontology that we presented in Chapter 5 to identify content-related questions appearing in MOOCs discussion forums. It accomplishes its task as a classification problem. This module processes discussion forum data to filter content-related questions. We proposed two feature indexing approaches to improve the accuracy of the results. Those indexing approaches depend on the subject ontology. Specifically, it leverage the concept hierarchy part of the subject ontology. This module also serves the question-answering module which we introduce in Chapter 7. The question-answering module assumes content-related questions input. As a result, it is important to extract those posts from MOOCs discussion forums. In this chapter we present this module in details.

6.1 Introduction

The advent of MOOCs allows learners to interact with other peers through the MOOCs forums. These interactions, in form of short text posts, generate substantial volume of data which offer fertile source for researchers to analyse learners' interactions.

MOOCs forums belong to the computer supported collaborative learning (CSCL) category. It is an important pedagogical element in MOOCs settings [Glance et al., 2013]. Typically, MOOCs platforms tag the forums data. These tags represent subject main concepts. Learners can search forum discussions data using these tags which provide assistance for direct learners' collaboration. Learners' collaboration is another element of CSCL that contributes to the learning process [Collazos et al., 2014].

Learners use these forums for different purposes and play different roles. They ask questions about the course contents which reflect the cognitive needs for MOOCs learners, they answer questions related to the content, they ask general questions which reflects social needs of MOOCs learners; they answer general questions, and/or they add comments and suggestions [Ramesh et al., 2014]. The lack of any prerequisites for registering in MOOCs had led to a great figure of registrants in these courses. As a result, they extensively use MOOCs forums and generate substantial amount of data which contributes to the information overload problem [Gulatee and Nilsook, 2016]. Usually, MOOC discussion forums have a significant number of indistinguishable threads. It also doesn't offer a service to cluster related topics or to link related topics together [Onah et al., 2014b].

Analysing discussion forms allows researchers to rather understand students learning, offer effective feedback for students, and improve learners engagement which results in improving MOOCs retention rate; a major issue in MOOCs [Onah et al., 2014a, Onah et al., 2014b].

Automatic analysis and filtering of forums data mitigates the effects of the information overloading problem, since those registrants are able to get answers for their questions quickly without reading a large number of peer comments and answers. In this research, we aim to filter MOOCs discussion forums to offer effective automatic feedback/answers for students queries and questions in responses to their cognitive needs. In our research reported in this dissertation,

we propose a system to automatically answer learners' content-related questions [Shatnawi et al., 2014]. However, as aforementioned MOOCs forums data contain different categories. As a result, it is important to filter these posts to identify content-related questions appearing in those posts.

Forum discussions relatively consist of short text posts. Sparseness, diversity, massiveness, immediacy, and irregularity are the major characteristics of short text. Short text classification is negatively affected by these characteristics. Short length text typically tends to have poor informative content thus, it leads to weak linkage to certain topics. Moreover, one can express the same topic in totally different ways (diversity), reducing the possibility of a feature term's appearing in several different posts. As a result, short text classification based on feature term co-occurrence often has weak accuracy results [Wang et al., 2012, Liu et al., 2010, Song et al., 2014]. Although MOOCs forum discussions have most of these characteristics, their contents revolve around a subject topic. In this research, we utilise this property to overcome the sparseness feature which results in improving short text classifiers in MOOCs discussion forums.

Research studies affirm that MOOCs registrants suffer from information overload [Gulatee and Nilsook, 2016, Onah et al., 2014a, Onah et al., 2014b] due to the significant number of registrants who generate great amount of data through the use of MOOCs forums . Also, course facilitators suffer from the same problem and they are able to reply to small fraction of the registrants' posts. In this research we aim to answer the following questions:

- Can the short text classification approach support other services to enhance learners' experiences when adopted in MOOCs?
- What are the characteristics of MOOCs forums data? And how can we utilise these characteristics to classify these forums data?
- What is the effect of using ontology-based feature indexing on classifying MOOCs forum data?

Before delving to the empirical part of this research, it is important to give formal definitions to some terms such as concept hierarchy, concept-based indexing, and concept hierarchy indexing.

Definition 7. Concept Hierarchy

Let C $=\{c_1, c_2, ..., c_n\}$ be a set of concepts in a course; then
Concept hierarchy is a hierarchical tree structure, with a root and subtrees of children with a parent node, represented as a set of linked nodes. Each node value represents a subject concept $c_i \in C$.

Definition 8. Child is a node directly connected to another node when moving away from the Root.

Definition 9. Parent is the converse notion of a child.

Definition 10. Siblings are group of nodes with the same parent.

Definition 11. Ancestor is a node reachable by repeated proceeding from child to parent.

Definition 12. First Common Parent Let c $=\{c_1, c_2, ...c_m\}$ be a $\subset$ of C; then c$\prime$ is the first common parent $\Leftrightarrow$ $c\prime \in c$ and $c\prime$ is the root for the minimum subtree contains all $c_i \in c$

Definition 13. Level : the level of a node is defined by 1 + (the number of connections between the node and the root).

Definition 14. Concept-based indexing : is an orderless document representation- only the count of concepts mattered; where each concept is a node value in the concept hierarchy.

Definition 15. Hierarchical concept-based indexing : is a concept-based indexing where multiple concepts are replaced by their first common parent.

After we defined the terminologies which we use to describe this module. The following sections describe the short text classification module. We refer to these definition wherever we used it.

TABLE 6.1: Dataset Statistics

Label	Posts	Min characters	Max characters
Content related question	248	22	2732
Content related answer	211	9	3266
General question	56	7	883
General answer /comments	95	26	1203

6.2 Data Collection

We collected forum discussions data for a MOOC which was offered by Stanford University in 2013. The course is "Introduction to Database" and available in archived mode at https://class2go.stanford.edu/db/Winter2013. Learners in this course can initiate a new post (thread) which typically contains a question for the course facilitators or other learners or a reply to existing threads to answer a question or elaborate upon other learners' answers. The course has 3684 posts: 203 unread posts, 134 unanswered questions, and 829 unresolved follow up. For the sake of this research we avoided posts about technical issues, software installations and debugging, or those which contain only URLs for external resources as these posts beyond the scope of our research. As a result, we collected 610 posts. A research for classification MOOCs forums data proposed eight categories (labels) for the forums data. However, we used the inductive methodology, similar to the methodology in [Stump et al., 2013], to label the collected posts. As a result, we subsumes the posts categories into four classes which are: content-related question, content-related answer, general question, and general answer. Table 6.1 summarises the collected posts and their labels. We removed URLs and emotional symbols from these posts. Finally, we converted all posts to lower case letters and we removed punctuation marks. Also we anonymised these posts and removed the timestamps to adhere to the data usage agreement listed in the terms and conditions web page.

6.3 Feature Indexing

Text classification deals with a high-dimensional feature space. Basically, the feature space consists of all unique terms (words) which appear in a corpus. A feature space dimensionality is inversely proportional to classifiers performance [Yang and Pedersen, 1997, Song et al., 2014, Liu et al., 2010]. For that, reducing the feature space by eliminating noisy terms improves the efficiency and effectiveness of these classifiers. Short texts such as tweets, SMS, and MOOCs forum discussions are characterised by sparseness and diversity which make the traditional feature selection approaches ineffective and result in poor performance [Song et al., 2014, Wang et al., 2012]. As a result, we need a feature selection approach that reduces the feature space dimensionality and decreases the diversity of these features.

Ontology of a subject consists of the subject concepts and the relations among these concepts. The subject concepts are organised in a hierarchical structure. The root represents the subject itself. Each level consists of some of the subject concepts in parent-child relationships. Each concept has a set of properties. This ontology is a conceptual representation of the subject's knowledge in a formal representation (machine readable) format. In our research reported in this dissertation, we proposed a framework to automatically build a subject ontology [Shatnawi et al., 2014]. We used the proposed approach to build the subject ontology for the "introduction to database" course. In this research, we propose two ontology-based feature indexing approaches to substitute the traditional term/phrase based indexing. MOOCs forum discussions data revolve around subject concepts. As a result, we propose a concept-based indexing approach, which is described in Definition 14, to minimise the feature space dimensionality. Then, we proposed another feature selection approach. This approach is based on the concept hierarchy of a subject ontology, which is described in Definition 15. In this approach, we use a higher-level concept to

replace multiple terminal concepts or low-level concepts (higher level orthogonal dimensions), which is described in Definition 12,. The experimental results approve that both approaches enhance the classifiers performance, and the later approach outperforms the former one.

6.3.1 Unigram-based Indexing

In document frequency thresholding, the corpus is represented in two-dimensional array where a row represents a document (post) in the corpus and a column represents a term (word). A cell value represents the frequency of that word in a document [Harris, 1954]. A well known and commonly used approach for feature selection is to keep only those terms which have a frequency above a given threshold (document frequency thresholding). However, this approach is not effective in classifying short text including the MOOCs forum discussions which is the target domain for this research due to the sparseness and diversity characteristics of short text corpus [Yang and Pedersen, 1997]. Our experimental results are aligned to other research results of short text classification which proved that the accuracy of the well known classifiers is low.

6.3.2 Concept-based Indexing

MOOCs forum discussions typically revolve around subject concepts. We utilise this characteristic to propose a new indexing scheme instead of the original term indexing scheme, which is described in Definition 14,. We retrieved all concepts from the subject ontology. Then, we parse the document (post) to identify all concepts appearing in that document. Instead of having term-document matrix, we build a concept-document matrix. A concept may consist of multiple words (variable length phrases). This reduces the feature space

(diversity characteristic). Unlike n-gram indexing where all columns have n-gram phrases, we have variant-gram indexing. And unlike phrase indexing approach which primarily relies on natural language processing and part of speech (POS) tagging, our approach utilises the ontology structure to index the matrix.

6.3.3 Hierarchical Concept Indexing

To rather decrease the feature space, we utilise the concept hierarchy on the subject ontology. Instead of using terminal concepts or low level concepts, we substitute these concepts with their parents in higher levels, which is described in Definition 15,. In this approach, we parse the document and identify all concepts appear in that document. Then, we use the concept hierarchy to replace all these concepts by their closest common parent, which is described in Definition 12,.

6.4 Experimental Results and Analysis

We selected a number of text classifiers to test the effect of the feature indexing on these classifiers. We selected the top-performing, state of the art, classifiers in text classification [Yang and Pedersen, 1997] . We namely select Support Vector Machine (SVM), Neural Networks (NNet), Decision Trees (Tree), Random Forests (RF), Bootstrap Aggregation (BAGGING), and Supervised Latent Dirichlet Allocation (SLDA). We run these classifiers against the corpus using the aforementioned indexing approaches.

TABLE 6.2: Accuracy of the tested classifiers based on unigram indexing in binary classification settings

fold	SVM	SLDA	TREE	BAGGING	RF	NNET
1	0.522	0.578	0.479	0.643	0.535	0.341
2	0.441	0.461	0.432	0.609	0.403	0.514
3	0.507	0.477	0.486	0.705	0.438	0.429
4	0.529	0.5	0.466	0.679	0.459	0.508
5	0.6	0.447	0.535	0.667	0.521	0.52
6	0.48	0.421	0.588	0.732	0.429	0.5
7	0.507	0.419	0.514	0.667	0.515	0.567
8	0.512	0.469	0.585	0.661	0.508	0.446
9	0.513	0.44	0.597	0.672	0.468	0.548
10	0.469	0.355	0.46	0.681	0.592	0.567

6.4.1 Binary Classification Experiments

The first experiment examined the effect of the indexing approaches on binary classification for short text. The posts were manually labelled as content-related posts or non content-related posts. Next, we apply the classifiers on the corpus using the aforementioned three indexing approaches. Table 6.2 shows the results of the tested classifiers using the unigram indexing approach. We used k-fold validation with k=10. The best result achieved by the BAGGING classifier with accuracy of 73.2%. However, the remaining classifiers achieved close results with accuracy between 52% and 59.7%.

We repeated the same experiment using the proposed concept-based indexing approach. Table 6.3 shows the results of this experiment. Again, the BAGGING classifier achieved the best results with accuracy of 95.3%. It is clear that the performance of the BAGGING classifier has improved using the proposed indexing approach. All the remaining approaches, except the NNET classifier, achieved better results and have accuracy in the range from 83.3% - 89.9%.

Finally, we repeated the experiment using the second proposed concept-hierarchy indexing approach. Table 6.4 shows the results of these experiments. The best

TABLE 6.3: Accuracy of the tested classifiers based on the subject concept indexing in binary classification settings

fold	SVM	SLDA	TREE	BAGGING	RF	NNET
1	0.8	0.735	0.73	0.953	0.841	0.473
2	0.767	0.69	0.651	0.682	0.851	0.5
3	0.673	0.757	0.78	0.872	0.814	0.561
4	0.894	0.8	0.7	0.894	0.767	0.405
5	0.846	0.625	0.833	0.767	0.842	0.389
6	0.857	0.829	0.684	0.737	0.776	0.4
7	0.861	0.641	0.714	0.771	0.861	0.25
8	0.863	0.833	0.725	0.8	0.853	0.405
9	0.938	0.791	0.787	0.816	0.824	0.512
10	0.824	0.732	0.775	0.769	0.892	0.429

TABLE 6.4: Accuracy of the tested classifiers based on the subject concept hierarchy indexing in binary classification settings

fold	SVM	SLDA	TREE	BAGGING	RF	NNET
1	0.793	0.672	0.783	0.877	0.75	0.746
2	0.903	0.765	0.746	0.809	0.812	0.82
3	0.827	0.597	0.706	0.848	0.852	0.742
4	0.871	0.725	0.885	0.787	0.86	0.725
5	0.75	0.719	0.824	0.8	0.853	0.738
6	0.757	0.723	0.758	0.828	0.8	0.635
7	0.895	0.726	0.794	0.833	0.841	0.743
8	0.846	0.717	0.837	0.897	0.78	0.851
9	0.803	0.733	0.85	0.794	0.814	0.7
10	0.833	0.638	0.804	0.864	0.813	0.821

accuracy rate is achieved by the SVM classifier which was 90.3%. The BAGGING classifier achieved 89.7 % accuracy rate which is close to the BAGGING accuracy rate when using the concept indexing approach. However, at the macro-average level the BAGGING classifier didn't outperform its self under the concept-based indexing algorithm and so the SLDA classifier. On the other hand, the remaining classifiers achieved better accuracy rate using the concept-hierarchy indexing approach.

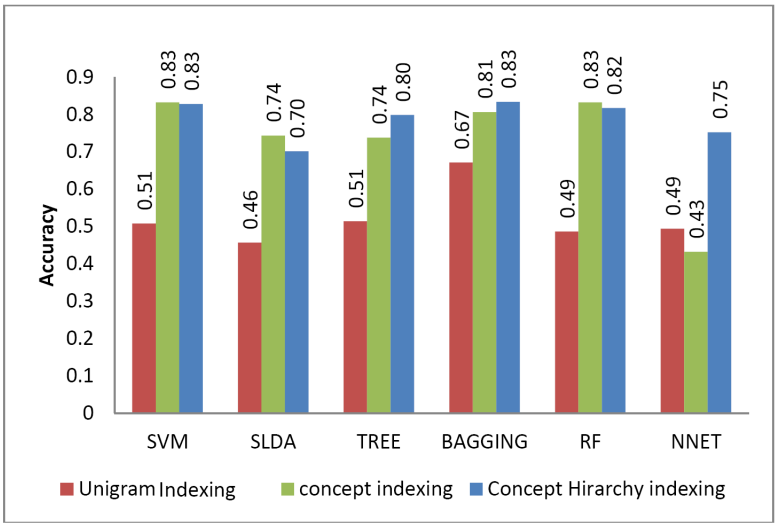


FIGURE 6.1: Macro-Average accuracy for the indexing approaches in binary classification settings

The experimental results shown in Tables 6.2, 6.3, and 6.4 support our claim that concept based indexing improves the classifiers accuracy for all the tested classifiers. And generally, the hierarchical indexing outperforms the concept indexing approach. We used k-fold validation with k=10. For each classifier we calculated the average accuracy of the 10 folds (macro-average). Figure 6.1 shows the macro-average accuracy for the tested classifiers against the indexing approaches. We found the macro-average for a classifier by taking the mean of all results over all folds as given in Equation 6.1 .

$$\mu = \frac{\Sigma_{k=1}^{10} Accuracy_k}{10} \tag{6.1}$$

6.4.2 Multi Class Classification

In this experiment, we use four classes to label the posts which are content-related question, content-related answer, general question, and general answer

TABLE 6.5: Accuracy of the tested classifiers based on unigram indexing in multiple-labels classification settings

fold	SVM	SLDA	TREE	BAGGING	RF	NNET
1	0.583	0.652	0.58	0.691	0.714	0.5
2	0.732	0.604	0.593	0.736	0.673	0.46
3	0.647	0.6	0.532	0.661	0.703	0.731
4	0.652	0.667	0.625	0.632	0.567	0.556
5	0.677	0.625	0.617	0.679	0.814	0.589
6	0.643	0.698	0.65	0.682	0.691	0.509
7	0.63	0.634	0.535	0.732	0.618	0.521
8	0.604	0.547	0.559	0.625	0.678	0.462
9	0.725	0.574	0.619	0.672	0.836	0.443
10	0.684	0.632	0.576	0.639	0.698	0.509

or comment. Then, we apply the classifiers using the tested indexing approaches. Table 6.5 shows the experimental results for the tested classifiers using the unigram indexing approach. The RF classifier achieved the best accuracy rate (83.6 %). However, most of the classifiers achieved close accuracy results in the range of 63% - 67%. These results are consistent with the binary classification results. However, the classifiers achieved better accuracy results in thye multi-label classification settings than the binary classification settings. On the other hand, the NNET classifier achieved better results in the binary classification settings.

Table 6.6 shows the experimental results for the concept-based indexing approach. The results in this experiment are consistent with the results in the previous experiment. The RF classifier achieved the best accuracy results (85.1%). The accuracy results for all classifiers in this experiment are better than the accuracy results of the unigram indexing approach. Also, the NNET classifier has the worst accuracy result.

Finally, we tested the classifiers using the concept-hierarchy based indexing approach. In this experiment, the RF classifier achieved the best accuracy rate (77.8%). However, this rate is worse than its best accuracy rate using the

TABLE 6.6: Accuracy of the tested classifiers based on the subject concept indexing in multiple-labels classification settings

fold	SVM	SLDA	TREE	BAGGING	RF	NNET
1	0.632	0.667	0.627	0.739	0.612	0.509
2	0.722	0.563	0.589	0.679	0.729	0.519
3	0.75	0.549	0.576	0.623	0.674	0.45
4	0.69	0.633	0.707	0.8	0.698	0.5
5	0.63	0.614	0.532	0.66	0.633	0.491
6	0.574	0.538	0.685	0.672	0.694	0.411
7	0.684	0.627	0.613	0.593	0.706	0.391
8	0.644	0.545	0.696	0.704	0.719	0.462
9	0.707	0.611	0.585	0.692	0.851	0.519
10	0.627	0.679	0.569	0.652	0.722	0.327

TABLE 6.7: Accuracy of the tested classifiers based on the subject concept hierarchy indexing in multiple-labels classification settings

fold	SVM	SLDA	TREE	BAGGING	RF	NNET
1	0.667	0.633	0.633	0.645	0.566	0.529
2	0.633	0.623	0.618	0.565	0.578	0.5
3	0.652	0.532	0.655	0.627	0.692	0.472
4	0.642	0.627	0.541	0.583	0.656	0.618
5	0.585	0.623	0.625	0.7	0.759	0.37
6	0.64	0.585	0.525	0.717	0.778	0.438
7	0.582	0.647	0.491	0.596	0.741	0.308
8	0.633	0.66	0.577	0.698	0.707	0.422
9	0.7	0.576	0.533	0.727	0.579	0.446
10	0.561	0.574	0.519	0.644	0.661	0.383

concept-based indexing approach. Nevertheless, the macro-average accuracy is better than the macro-average accuracy using the concept-based indexing approach. In this experiment, the NNET classifier achieved worse accuracy rate than the concept-based indexing approach experiment. Table 6.7 shows the experimental results.

Although the concept indexing approach improves the classifiers accuracy and generally the hierarchical indexing outperforms the concept indexing which is

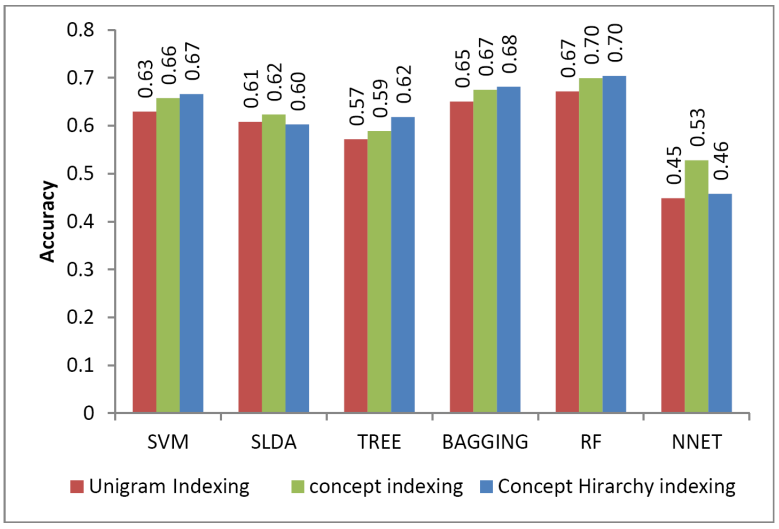


FIGURE 6.2: Macro-Average accuracy for the indexing approaches in multiple classification settings

aligned to the binary classification results, the accuracy improvement is small. Figure 6.2 shows the macro-average accuracy for the tested classifiers against the indexing approaches. In both experiments we use k-fold cross validation approach (k=10).

6.5 Discussion and Analysis

We examined the effect of using term/phrase feature indexing vs ontology based feature indexing on short text classification. We compiled a corpus form a MOOC forum discussions data. Then, we used a set of top performing classifiers to capture the effect of these indexing approaches. We ran our experiments based on the document frequency thresholding method for binary and multilabels classification settings. In the binary classification settings the two proposed indexing approaches improved the accuracy of the tested classifiers. Table 6.8 shows the improvement rate of these classifiers for the two proposed indexing approaches. The RF classifier achieved the best improvement rate

TABLE 6.8: Accuracy Improvement of the Proposed Indexing Approaches vs Unigram Indexing in Binary Classification Settings

Classifier	Indexing Approach	
	Concept Based %	Hierarchy Based %
SVM	32	32
SLDA	28	24
TREE	23	29
BAGGING	14	16
RF	34	33
NNET	-6	26

which is 34% in the binary classification setting using the concept-based indexing approach. However, the accuracy rate for the NNET classifier achieved was declined by 6%. On the other hand, using the concept hierarchy based indexing approach improved the accuracy for all classifiers. Although, the RF classifier achieved the best accuracy rate improvement. The NNET classifier achieved a significant improvement rate of 26%.

In the multi-label classification settings, the proposed indexing approaches slightly improved the accuracy of the tested classifiers. The best improvement occurred to the NNET classifier (8%). The remaining classifiers achieved accuracy improvements from 1 to 3%. The results are consistent with the results in the binary classification settings. The RF classifier achieved the best accuracy results (70%). Also, the hierarchical concept based indexing approach achieved better results than the concept based indexing approach.

In the binary classification experiment, the unigram indexing approach achieved the worst accuracy for the all used classifiers. Both the concept based indexing and the hierarchical concept indexing approaches achieved promising results and improved the accuracy of the tested classifiers. An exception of that was the Neural Network classifier (NNet) where the unigram indexing approach slightly outperforms the concept based indexing approach.

TABLE 6.9: Accuracy Improvement of the Proposed Indexing Approaches vs Unigram Indexing in Multi-labels Classification Settings

Classifier	Indexing Approach	
	Concept Based (%)	Hierarchy Based (%)
SVM	3	4
SLDA	1	-1
TREE	2	5
BAGGING	2	3
RF	3	3
NNET	8	1

The random tree forests and the BAGGING classifiers achieved the best performance 83% on the proposed ontology based indexing approaches. On the other hand, the multiple-labels classification results are aligned with the binary results in term of improving the classifiers accuracy. However, we have small accuracy improvements.

Based on our experiments, we recommend the use of subject ontology indexing approaches to classify MOOCs forums discussions. Also, for the sake of identifying content-related questions. It is recommended to filter content-related questions in two phases instead of using multi-label classification settings. First, we use the binary classification settings to identify the content-related posts (questions and answers). Then, we use the binary classification settings again to filter content-related questions. In future work we will explore more techniques to enhance the accuracy of the classifiers for short text in MOOCs settings. Also, we will compare the proposed indexing approaches to other feature selection techniques.

6.6 Summary

In this research, we tested the effect of two novel feature indexing approaches on classifying short text in MOOCs discussion forums. The proposed feature indexing approaches leverage the concept hierarchy of a subject ontology. These feature indexing approaches are concept-based indexing and hierarchical concept-based indexing. We tested these approaches in both binary classification and multi-class classification settings. Both approaches improve the accuracy of the test classifiers. As a result, these approaches can identify content-related questions in MOOCs discussion forums. The results of this research support the question-answering system which is presented in Chapter 7. The question-answering system offer answers to content-related questions. Consequently, it is important to filter MOOCs discussion forum posts to identify these posts.

Chapter 7

Question Answering Module

This chapter describes the third module of the proposed framework that is described in Chapter 4. This module leverages the output of the other framework modules which we described in Chapter 5 and 6. This modules takes content-related questions from the short text classification module which is described in Chapter 6, then, it queries the resulting subject ontology which is described in Chapter 5. Finally, it returns answers to these questions.

7.1 Introduction

As aforementioned earlier, registrants in MOOCs receive insufficient feedback to fulfil their cognitive needs [Ramesh et al., 2014]. So, this module aims to return answers to students' questions to fulfil their cognitive needs. Usually, learners use discussion forums to ask questions. This module automatically answers content-related questions. Generally, a question answering system consists of three components which are question classification, information retrieval, and answer extraction. As a result, the proposed question answering system consists

of three components which are the question analysis component, the question formation and ontology querying component, and the answer selection and aggregation component. Another function for this module is to automatically label learners' posts to give feedback for course facilitators about topics that are being asked for by learners. This enables course facilitators to get clues for topics that need more explanations or more resources to make these topics clear. As a result, they improve the quality of following course runs. We created an ontology for a subject manually in order to test the question-answering system. Then, we replaced that ontology by the resulting automatic subject ontology which we described in Chapter 5. The following steps describes the question-answering system steps and Figure 7.1 visualises this module.

1. Read content-related questions: this step accepts the question and split word by word.

2. Identify the subject terms in the post : in this step, the state table, which was created as a result of the DFA component in the subject ontology learning module which is described in Chapter 5, derives the process of identifying the subject terms appearing in the post. It works in similar way of compilers when parsing computer programs. Keep in mind that this state table is created with assistant of "Wordnet" API to include all synonyms of the subject terms.

3. Identify the properties for these terms: in analog way to the previous step the state table derives the process of identifying term properties.

4. Construct ontology queries: in this step the identified terms and their properties are converted to the ontology query format (ontology triples).

5. Generate feedback by aggragating the queries' results:finally, this step form the answer and present it to learners.

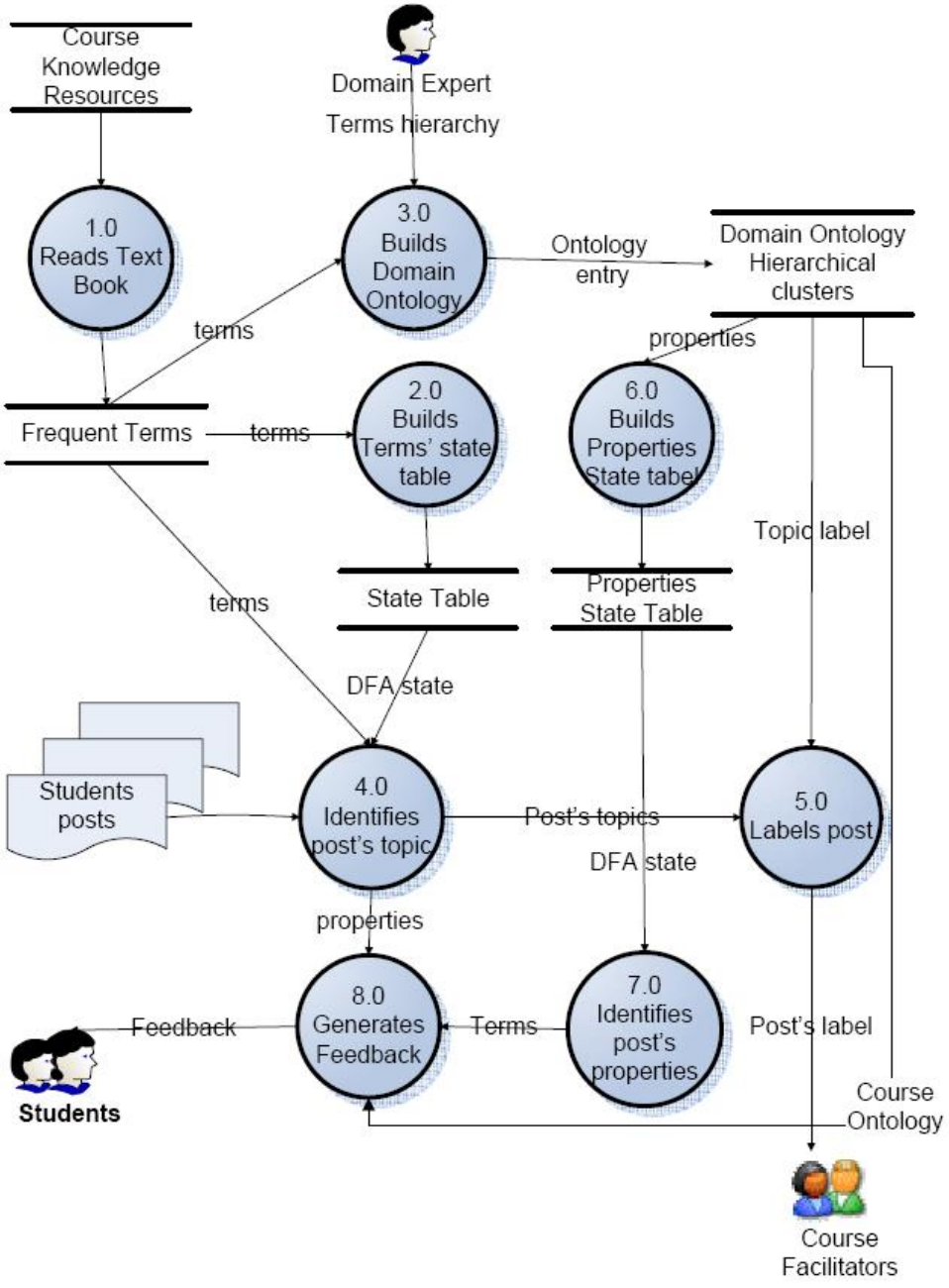


FIGURE 7.1: Students Posts Labelling and Feedback System.

7.2 The Question Answering Components

The first component is the question analysis component. This component aims to understand questions. It uses morph-syntactic analysis. First, it parses each question and represents it in a word-vector structure. Next, it identifies concepts and properties mentioned in the processed question. To do that, the state table component drives this process. Here it is worth mentioning that the efficiency and completeness of the subject ontology play an important role in the efficiency and accuracy of the obtained concepts and properties. The more representative the subject ontology is, the more accurate this component is. Consequently, the quality of the obtained answers is proportionally correlated to the quality of this component. As a result, one can use the question-answering module to assess the underlying subject ontology. We used the question-answering module to validate the applicability of the resulting subject ontology detailed in Chapter5.

The second component is the question formation and ontology querying Component. This component takes the output of the question analysis component and converts it into a set of triple patterns. Next, it uses these triple patterns to query the subject ontology. Each triple pattern query returns a statement. A statement is a combination of a resource (a subject concept), a property, and a property value (feedback/answer). The difference between an ontology query and a standard SQL query is that an ontology query allows implicit fact retrieval. For example, a triple pattern consists of "Difference(subject1, subject2)" is obtained implicitly from a rule over "characteristic (subject1) " and "characteristic(subject2)". To clarify that consider the following question: What are the differences between delete and drop commands in SQL?
We have two concepts here which are "delete" and "drop" commands. Also, we have a property which is"differences". Now, the system will form a query such as

"differenc(delete, drop)". However, "difference" is not one of the properties attached to any concept in the subject ontology. Instead, "difference" is a rule embedded in the question-answering system and the "differenc(delete, drop)" query is translated to multiple queries such as "syntax(delete)" and "syntax(drop)" for example. Then, the system queries the subject ontology to retrieve the answer parts.

The last component is the answer compiling component. This module takes the results of the triple patterns queries and aggregates these results to form a single answer for learners' questions. Due to the representation of a subject knowledge as an ontology, the subject knowledge is made explicit. This ensures that an answer component follows well formed standards and have profound effect on students learning. As a result, this module compiles these components and presents it to MOOCs registrants.

7.3 Experimental Setting and Results

In our experimental work, we used the 2013 version of "Introduction to Database" course, offered by *Coursera* [1]. we played the role of domain expert to build the subject ontology. Then, for every subject-concept we assigned a set of properties. We assigned an answer (feedback) for every property. For example "select command" is a concept in the database subject. This concept has a "syntax", "example", and "purpose" properties. Each property has a feedback value. a "syntax" property may have the following feedback "select [field|function] from table list ...". The system retrieves this answer in result of the triple pattern query (select,syntax,feedback). Finally, this answer is sent back to students. The following is an example that clarifies the domain ontology that we created to test our approach.

[1] www.courseara.org

Course Ontology Example 1. $\Theta := (T, P, C^*, H, Root)$
T : "key","primary key","data", "information","database management system","foreign key","relationship", "conceptual model",....
P: "definition","type", "syntax", "use", "advantage",....
C^*: "relationship *is a* conceptual model" , "schema *consists of* attributes" , "foreign key is part of relationship",...
H: Concept hierarchy parent(DBMS,RDBMS), Parent(RDBMS,Table),...
Root: Database.

Typically a knowledge-base repository serves as an input for a typical question-answering system. We configured the subject ontology to serve as a knowledge-base for the proposed question-answering system which allows semantic reasoning to answer questions. We used a list of predefined properties in the configuration process. In the education-content space, four types of properties were suggested, which are: definition, synonyms, example, and further explanation [Boyce and Pahl, 2007]. We extended these properties to represent in more details the underlying subject knowledge by adding the following properties: purpose, syntax, characteristic, advantage, and disadvantage. We extended these properties to cover the subject knowledge ("Introduction to Database"), these properties remain valid for IT courses. We used "Wordnet" synonyms to syntactically extend the property list.

The properties were attached to the concepts in the ontology, and each (concept, property) pair was assigned a corresponding feedback, i.e. an answer to a question containing a concept and its property. Consequently, the knowledge-base for the answering system is represented as (concept, property, feedback) triples. Fig. 7.2 is a compact Web Ontology Language (OWL) code that represents an example of the ontology triple structure for the "dbms" concept and its "definition" property.

```
<Class rdf:ID="database" />
<Class rdf:ID="concept" />
<Class rdf:ID="property" />
<Class rdf:ID="DBMS" >
<rdfs:subClassOf rdf:resource="database" />
</Class>
<owl:ObjectProperty owl:name="definition">
  <owl:domain owl:class="DBMS" />
  <feedback> is a computer software application that interacts with
    the user, other applications, and the database itself to capture
    and analyze data. </feedback>
</owl:ObjectProperty>
</rdf:RDF>
```

FIGURE 7.2: OWL Code Snip

We then prepared a collection of questions which we use to test our system. The test collection was collected from database management textbooks and from database forums (learners questions) [2][3][4]. For every post, we store a label and an answer key. Then we run our system to assign a label and provide feedback for every post. We used precision, recall, and F-measure to validate the results our system. We used semantic text similarity based on latent semantic analysis using the SIMILAR tool [Rus et al., 2013] to evaluate the relevance of retrieved answer to the stored answer key. More details about the latent semantic analysis is described in Chapter 2. The Equations 7.1, 7.2, and 7.3 are used to validate the system.

$$Precision = \frac{A}{A+B} \tag{7.1}$$

$$Recall = \frac{A}{A+C} \tag{7.2}$$

[2]Database Systems: Applicational Approach to Design, Implementation, and Management; 4th Edition

[3]Database Management Systems; 2nd Edition.

[4]Fundamentals of Database Systems;6th Edition

$$F\text{-}measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{7.3}$$

Where A is the number of correct labels obtained, B is the number posts that was not labelled and C is the number of incorrect labels retrieved. Table 7.1 shows the experimental results of the system. The results show the potential of the system in providing the students with timely feedback. The system achieved promising results in term of precision, recall, and *F-measure* as shown in Table 7.1. However, for some posts the system failed to label the post, consequently it failed to retrieve any feedback. A possible reason behind that is the lack of domain knowledge where the posts were about technical issues related to the database system or about contents not related to the database management system. In some other cases, however, the system was able to successfully label the post at the time it failed in retrieving a relevant feedback. Some posts have multiple topics and properties; as a result the system retrieved extra feedback which is not relevant to the post. A possible solution for that is using part of speech tagging and divide the post into multiple statements.

TABLE 7.1: Experimental results

	LABELLING (%)	FEEDBACK (%)
Recall	82	72
Precision	91	84
F-measure	86	78

Finally, we replaced the subject ontology by an automatic generated subject ontology. First, we used the subject ontology generated by the subject ontology learning module which is presented in Chapter 5. Second, we used Text2Onto tool [Cimiano and Völker, 2005] to generate a subject ontology. In this step, we aimed to measure the effect of using different ontologies on the efficiency of the proposed question-answering system. Table 7.2 shows the results of our experiments. It is clear that the subject ontology component has a major effect on the quality of the answers generated by the question-answering system.

TABLE 7.2: The Effect of Subject Ontologies on The Proposed Question-Answering System

	Manual Created Ontology	Resulting Ontology	Text2Onto Ontology
Recal	72	67.9	71.4
Precision	84	72.6	22.2
F-Measure	78	70	33.9

The best performance for the system was with the manually created subject-ontology, which is expected, however, our proposed ontology learning module generated a subject ontology which allows the question-answering system to achieve close results to the manually created ontology results (78% vs 70%).

7.4 Discussion and Analysis

Usually a typical question answering system consists of three tasks, namely, question processing, document parsing, and answer processing. However, the research in this chapter proposes a question answering system based on semantic analysis. Semantic analysis based question answering systems use a knowledge base to answer questions. In this research, we used subject ontologies as a knowledge base. Instead of using information retrieval techniques, we used semantic reasoning to find answers for users' questions. First, the system classifies (labels) questions by leveraging the subject concept hierarchy. This is an important step and affects the quality of the returned answers. The system was able to correctly label 82% of the questions with 91% accuracy which is a good accuracy for the proposed question answering system. Question classification depends on the quality of the underlying subject ontology. As a result, this task gives indication for the quality of the subject ontology. The more accurately the system labels questions, the more reliable the underlying subject ontology is. The type of the questions that we used to test the proposed system falls in the lower four layers of Bloom's Taxonomy [Bloom, 1956]. Instead of using IR techniques to retrieve answers, we used semantic analysis (reasoning)

to form answers. This allows the system to answer questions beyond the "Wh" questions. Also, it can offer answers aligned to the educational standards. The system was able to answer 67.9 % of the tested questions using the automatically generated subject ontology. Although this ratio seems to be low, it is close to the accuracy achieved by using the manually created subject ontology (72%). And it is much better than the accuracy achieved by the automatically generated ontology by the Text2Onto tool (22.2 %). In spite of the fact that the accuracy of answers depends on both the rules of the ontology and the concept hierarchy, given that the rules are fixed for the three ontologies then the concept hierarchy for the automated subject ontology resulted form our proposed system in Chapter 5 is reliable.

7.5 Summary

Domain ontology and NLP techniques can scaffold teaching and learning processes in MOOCs settings. Domain ontology is an effective representation of course content knowledge. We proposed a feedback system for MOOCs settings. Our system represents a MOOC's contents using domain ontology notations. We separated the knowledge part from the processing part. As a result, the system capable of learning new knowledge without changing the processing part. We also generated deterministic finite automata using natural language expressions derived from domain ontology instances. We created simple tools to automate and mange domain ontology population. We used a manually created subject ontology and an automated one.

Chapter 8

Conclusion and Future Directions

MOOCs open up new horizons for education. They can reach students regardless of their geographical presence. However, the massiveness feature of these courses complicated the existing information overloading problem. As a result, many registrants left these courses at early stages. High dropout ratio is a salient feature of all MOOCs and threat the continuity and efficiency of these courses. Hence, it is important to enhance the quality of the offered services to mitigate the effect of the massiveness feature. Discussion forums are one of the most important pedagogical elements used in MOOCs.

Offering timely feedback for registrants queries, especially content-related queries, is one of the service-enhanced techniques to support MOOCs. Subject ontologies are one of the available tools to support technology enhanced learning systems. However, subject ontology learning is a complex and time consuming task. The available ontology learning tools are not appropriate for educators, even those who have some IT skills.

The research reported in the last seven chapters of this dissertation aimed to support MOOCs by offering automatic feedback for content-related questions. It aimed to build a subject ontology using textual learning objects. Then, it

aims to leverage this subject ontology to support a question-answering system. The question-answering system answers content-related questions that learners ask in MOOC discussion forums. Students use discussion forums for different purposes. As a result, it is important to filter content related questions in order to answer these questions. Filtering content-related questions is one of our objectives in this research.

8.1 Conclusion and Reflection

The research reported in this dissertation answers the following research questions:

- How can we represent a subject knowledge to support automatic feedback? Ontologies can represent a subject knowledge. They make the knowledge of a subject, that is distributed among different learning objects, explicit. Textual learning objects are possible and appropriate source for building a subject ontology.

  Data mining techniques and NLP tools can support the ontology learning process especially for technology enhanced learning. We used NLP tools to discover terms and concepts embedded in subject learning objects. Then, we used the FP-Tree and FP growth algorithms in a novel approach to support the concept-hierarchy construction for a subject ontology. We developed a general deterministic finite automata for the subject concepts. Then, we used it to build a transactional database for the FP-Tree algorithm. After that, we customised the FP-Tree to adhere to the concept hierarchy requirements. Finally, we used a heuristic function derived by the FP growth algorithm to enhance the quality of the subject ontology. We achieved promising results for developing subject ontologies. A comparative validation approach proved that the quality of

the resulting ontology is close to the quality of a subject ontology created by the subject experts. Moreover, the resulting ontology is much better than the ontology which was generated by the Text2Onto tool when it was embedded in a question answering system in the MOOCs settings.

Consequently, our research showed that, in the educational context, it is possible to develop subject ontologies by leveraging general NLP techniques with the assistance of the proposed general natural language deterministic finite automata, the FP-Tree algorithm, and the FP-Tree growth algorithm. These techniques are subject independent which it makes the proposed approach applicable to different subject domains.

- How can the knowledge representation underpin automatic feedback? We extended the resulting subject ontology to serve as a knowledge resource for the question-answering system. We achieved that by attaching a "feedback" property to each (concept,property) pair in the resulting ontology. To answer content-related questions the system uses the state table, which is generated by the DFA component in the ontology learning phase, to parse these questions. As a result, it identifies all concepts and their properties appearing in these questions. Next, it applies a set of rules to form ontology queries in triple format. Finally, it queries the subject ontology to retrieve the "feedback" property values and presents it to the learners. We used end of chapter questions which have well defined educational standards to validate the system. Experts evaluated the correctness of the returned answers. Then, we used expert evaluations to select the best similarity measure appropriate to automatically compare the returned answers with the answers provided by the book authors. We found that LSA-based text similarity techniques gave the closest scores in comparrison to the experts' scores. The system is able to answer a significant portion of the tested questions whenever the underlying subject ontology comprehensively covers the subject knowledge. Also, the system

labels these questions to give course facilitators feedback about the most frequent topics appearing in these questions. In collaborative learning settings, a subject ontology underpins question-answering systems to automatically answer learners' questions which mitigates the effects of the information overloading problem. These systems can support e-learning systems with a great number of registrants such as MOOCs.

- How can the knowledge representation underpin MOOCs discussion forums analysis? Subject ontologies are useful for short text classification and topic detection. We proposed two novel feature indexing approaches for short text classification. These approaches leverage the concept hierarchy of subject ontologies. The first is the concept-based indexing approach which represents short text in term of subject concepts appearing in these texts. Instead of using unigram indexing or phrase indexing, it uses the state table component to parse short texts, then it identifies the subject concepts and uses these concepts for feature indexing. The other approach is the hierarchical concept-based approach which aims to reduce the feature space by using concepts which appear in higher levels instead of terminal or low level concepts. Both approaches achieved promising results when they were used with the state of the art classifiers on short texts. They significantly improved the accuracy of the tested classifiers for binary classification and multi-class classification settings. We ran our experiments in the context of MOOCs settings to identify those posts which contain content related questions. However, these approaches are applicable for short text classification across different domains.

8.2 Summary of Contributions

The current MOOC settings expanded the traditional classroom settings in term of the number of learners in a class. However, it didn't expand the feedback

element to the same level. The proposed feedback module aims to expand the feedback element to suit the volume of learners in MOOCs settings. The framework consists of three main modules to achieve the research objectives. These modules are subject ontology learning, question-answering, and short text classification.

The subject ontology learning module employed a data mining-based technique to construct the concept hierarchy for the identified concepts. A heuristic function based on concept association mining drived the concept hierarchy-construction module to enhance the quality of the concept hierarchy structure by resolving multiple occurrences within the hierarchy, and by solving the siblings problem. The DFA representation and the concept-hierarchy construction modules make our approach applicable to different subjects.

The question-answering module answers content-related questions. It receives users' questions (content-related), analyses each question to identify subject concepts and the specific attributes appearing in that question. Next, it queries the resulting subject ontology to get the answer components. Then, it synthesises the answer and presents it to the users. The quality of the resulting subject ontology has a profound effect on the maturity and accuracy of the returned answers. If the ontology fails to capture some portions of the knowledge then the question-answering module is not able to answer any question related to these portions.

Another area for the resulting subject ontology is to support discussion forums analysis. We proposed novel ontology driven feature indexing approaches for classifying short text documents. Bothe approaches enhanced the accuracy of the test classifiers. We employed these approaches for filtering MOOCs discussion forums to capture those posts that contains content-related questions.

8.3 Future Directions

- The quality of subject ontologies plays a vital role in the quality of the answers generated by the question-answering system. Hence, we intend to extend our research to allow instructors or even learners to edit the subject ontology part through a well designed user friendly interface suitable for the users in the educational field. Ontology editing has a two-fold value. First, it allows instructors/learners to add any missing concepts that have not been captured by the proposed system; as a result, it enhances the quality of the subject ontology which in turn improves the accuracy of the question-answering system. Second, it builds consensus for the subject ontology which is an important part of the ontology definition and cannot be achieved without having multiple perspectives reflected in the subject ontology. In this direction we have to offer an ontology editor tool for educators that hide the complexity of the ontology structure.

- It is possible to connect subject ontologies to quality assurance ontologies in order to generate answers according to well defined educational standards. The quality assurance ontology can have specific rules to synthesise answers for questions in the higher levels of Bloom's taxonomy. We plan to test the applicability of the current proposed subject ontology learning approach on the quality assurance ontology.

- Another avenue for a future research is to examine the effectiveness of question-answering systems in MOOCs settings. Specifically, what an effect this service does have on the dropout ratio. This requires to put the proposed framework in action and extend it to include a dialogue system with learners. Also, quantify this effect through a voting system for the offered answers. In addition, the subject ontology can be connected to other learning objects such as images, audio, or video objects to offer

feedback beyond the textual answer that currently exists. This work requires to build other ontologies and to connect these ontologies together.

- Offering adaptive learning is one of the goals of online learning management systems. Leveraging the short text classification research and building user model ontologies to offer adaptive user-based learning is a feasible research track. Also, it is useful to explore clustering MOOCs discussion forums techniques to group similar posts together and to fulfil learners' cognitive needs and social needs.

- Students' engagement can be easily noticed in face-to-face teaching style. However, the current MOOCs settings can't measure students' engagement. So, applying short text classification for sentiment analysis in MOOCs discussion forums enables course facilitators to get both qualitative and quantitative feedback about students learning and engagement. This can lead to more services to keep MOOCs registrants up to the finish line.

Bibliography

Aggarwal, C. and Zhai, C. (2012a). A survey of text clustering algorithms. In Aggarwal, C. C. and Zhai, C., editors, *Mining Text Data*, pages 77–128. Springer US.

Aggarwal, C. C. (2011). Social network data analytics, chapter an introduction to social network data analytics. *IBM TJ Watson Research Center Hawthorne, NY 10532*, 13.

Aggarwal, C. C. (2012). Mining text streams. In *Mining Text Data*, pages 1–10. Springer Science & Business Media.

Aggarwal, C. C. and Han, J. (2014). *Frequent pattern mining*. Springer.

Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003). A framework for clustering evolving data streams. In *In VLDB*, pages 81–92.

Aggarwal, C. C. and Zhai, C., editors (2012b). *Mining Text Data*. Springer.

Agrawal, R., Golshan, B., and Papalexakis, E. E. (2016). Toward data-driven design of educational courses: A feasibility study. In Barnes, T., Chi, M., and Feng, M., editors, *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, page 6. International Educational Data Mining Society (IEDMS).

Agrawal, R., Imielinski, T., and Swami, A. (1993). Database mining: A performance perspective. *Knowledge and Data Engineering, IEEE Transactions on*, 5(6):914–925.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ahmed, K. B. S., Toumouh, A., and Malki, M. (2012). Effective ontology learning: Concepts' hierarchy building using plain text wikipedia. In *ICWIT*, pages 170–178. Citeseer.

Al-Yahya, M., George, R., and Alfaries, A. (2015). Ontologies in e-learning: review of the literature. *International Journal of Software Engineering and Its Applications*, 9(2):67–84.

Allan, J., Harding, S., Fisher, D., Bolivar, A., Guzman-Lara, S., and Amstutz, P. (2005). Taking topic detection from evaluation to practice. In *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04*, HICSS '05, pages 101–110, Washington, DC, USA. IEEE Computer Society.

Arai, K. and Handayani, A. N. (2012). Question answering system for an effective collaborative learning. *IJACSA Journal*, 3(1).

Aroyo, L. and Dicheva, D. (2004). The new challenges for e-learning: The educational semantic web. *Educational Technology & Society*, 7(4):59–69.

Aslam, J., Pelekhov, E., and Rus, D. (2006). The star clustering algorithm for information organization. In Kogan, J., Nicholas, C., and Teboulle, M., editors, *Grouping Multidimensional Data*, pages 1–23. Springer Berlin Heidelberg.

Augustson, J. and Minker, J. (1970). An analysis of some graph theoretical cluster techniques. *Journal of the ACM (JACM)*, 17(4):571–588.

Beil, F., Ester, M., and Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 436–442, New York, NY, USA. ACM.

Benamara, F. (2004). Cooperative question answering in restricted domains: the webcoop experiment. In *Proceedings of the Workshop Question Answering in Restricted Domains, within ACL*. Citeseer.

Biemann, C. (2005). Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93.

Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals.

Boyce, S. and Pahl, C. (2007). Developing domain ontologies for course content. *Educational Technology & Society*, 10(3):275–288.

Buitelaar, P., Cimiano, P., and Magnini, B. (2005). Ontology learning from text: An overview. In *Ontology learning from text: methods, evaluation and applications*, pages 3–12. IOS press.

Carbonell, J., Harman, D., Hovy, E., Maiorano, S., Prange, J., and Sparck-Jones, K. (2000). Vision statement to guide research in question & answering (q&a) and text summarization. *Rapport technique, NIST*.

Chen, R.-C., Lee, Y.-C., and Pan, R.-H. (2006). Adding new concepts on the domain ontology based on semantic similarity. In *International Conference on Business and information*, pages 12–14. Citeseer.

Chi, Y.-L. (2009). Ontology-based curriculum content sequencing system with semantic rules. *Expert Systems with Applications*, 36(4):7838–7847.

Chowdhury, G. (2010). *Introduction to modern information retrieval*. Facet publishing.

Cimiano, P. (2006). Ontology learning from text. In *Ontology Learning and Population from Text*, pages 19–34. Springer US.

Cimiano, P., Mädche, A., Staab, S., and Völker, J. (2009). Ontology learning. In *Handbook on ontologies*, pages 245–267. Springer.

Cimiano, P. and Völker, J. (2005). Text2onto: A framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Natural Language Processing and Information Systems*, NLDB'05, pages 227–238, Berlin, Heidelberg. Springer-Verlag.

Coll, C., Rochera, M. J., and de Gispert, I. (2014). Supporting online collaborative learning in small groups: teacher feedback on learning content, academic task and social participation. *Computers & Education*, (0):–.

Collazos, C. A., González, C. S., and García, R. (2014). Computer supported collaborative moocs: Cscm. In *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*, IDEE '14, pages 28:28–28:32, New York, NY, USA. ACM.

Connolly, T. M. and Begg, C. (2001). *Database Systems: A Practical Approach to Design, Implementation, and Management*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition.

Cooper, S. and Sahami, M. (2013). Reflections on stanford's moocs. *Commun. ACM*, 56(2):28–30.

Corley, C. and Mihalcea, R. (2005). Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18. Association for Computational Linguistics.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Coursera (2013). Coursera piazza report for db course. `https://piazza.com/stats/report/hbtmlzostxhfc`. Accessed 31 08 2013.

Crowley, R. S. and Medvedeva, O. (2006). An intelligent tutoring system for visual classification problem solving. *Artif. Intell. Med.*, 36(1):85–117.

Cui, Y. and Wise, A. F. (2015). Identifying content-related threads in mooc discussion forums. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, pages 299–303, New York, NY, USA. ACM.

Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92', pages 318–329, New York, NY, USA. ACM.

Davies, M. (2008-). The corpus of contemporary american english: 450 million words, 1990-present.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 269–274, New York, NY, USA. ACM.

Dicheva, D. and Dichev, C. (2006). Tm4l: Creating and browsing educational topic maps. *British Journal of Educational Technology*, 37(3):391–404.

Dicheva, D., Sosnovsky, S., Gavrilova, T., and Brusilovsky, P. (2005). Ontological web portal for educational ontologies. In *Proc. Of Applications of Semantic Web Technologies for E-Learning Workshop (SW-EL 05) in conjunction with 12th Int. Conf. on Artificial Intelligence in Education (AI-ED 05), Amsterdam*, pages 19–29.

Ezen-Can, A. and Boyer, K. E. (2013). Unsupervised classification of student dialogue acts with query-likelihood clustering. In *EDM*, pages 20–27.

Feinerer, I. and Hornik, K. (2015a). *tm: Text Mining Package*. R package version 0.6-1.

Feinerer, I. and Hornik, K. (2015b). *wordnet: WordNet Interface*. R package version 0.1-10.

Feng, D., Shaw, E., Kim, J., and Hovy, E. (2006). An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, pages 171–177, New York, NY, USA. ACM.

Fiedler, A. and Tsovaltzi, D. (2003). Automating hinting in an intelligent tutorial dialog system for mathematics. *IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, pages 23–35.

Freitag, D. and McCallum, A. (2000). Information extraction with hmm structures learned by stochastic optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 584–589. AAAI Press.

Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S. (2005). Mining data streams: a review. *SIGMOD Rec.*, 34(2):18–26.

Glance, D. G., Forsey, M., and Riley, M. (2013). The pedagogical foundations of massive open online courses. *First Monday*, 18(5).

Goodman, B., Soller, A., Linton, F., and Gaimari, R. (1997). Encouraging student reflection and articulation using a learning companion. In *Proceedings of the AI-ED 97 World Conference on Artificial Intelligence in Education*, pages 151–158.

Gravetter, F. J. and Forzano, L.-A. B. (2015). *Research Methods for the Behavioral Sciences*. CENGAGE Learning, 200 Firsr Stamford Place, 4th Floor Stamford, CT 06902 USA, 5th edition edition.

Gulatee, Y. and Nilsook, P. (2016). Mooc's barriers and enables. *International Journal of Information and Education Technology*, 6:826–830.

Gupta, S., Mittal, S., and Mittal, A. (2008). Eureqa: Overcoming the digital divide through a multidocument qa system for e-learning. In *The National Conference on emerging trends in Information Technology*.

Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2):1–12.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hatala, M., Gasevic, D., Siadaty, M., Jovanovic, J., and Torniai, C. (2012). Ontology extraction tools: An empirical study with educators. *Learning Technologies, IEEE Transactions on*, 5(3):275–289.

He, Q., Chang, K., Lim, E.-P., and Banerjee, A. (2010). Keep it simple with time: A reexamination of probabilistic topic detection models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1795 –1808.

Henze, N., Dolog, P., and Nejdl, W. (2004). Reasoning and ontologies for personalized e-learning in the semantic web. *Journal of Educational Technology & Society*, 7(4):82–97.

Hermjakob, E., Hovy, U., Gerber, L., Junk, M., and Lin, C.-Y. (2000). Question answering in webclopedia. In *Proc. of the TREC-9 Conference, NIST, Gaithersburg, MD*.

Hirschman, L. and Gaizauskas, R. (2001). Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

Hornik, K. (2014). *openNLP: Apache OpenNLP Tools Interface*. R package version 0.2-3.

Hornik, K., Buchta, C., and Zeileis, A. (2009). Open-source machine learning: R meets Weka. *Computational Statistics*, 24(2):225–232.

Hotho, A., Maedche, A., and Staab, S. (2002). Ontology-based text document clustering. *KI*, 16(4):48–54.

Hyman, P. (2012). In the year of disruptive education. *Commun. ACM*, 55(12):20–22.

Isotani, S., Mizoguchi, R., Isotani, S., Capeli, O. M., Isotani, N., de Albuquerque, A. R. L., Bittencourt, I. I., and Jaques, P. (2013). A semantic web-based authoring tool to facilitate the planning of collaborative learning scenarios compliant with learning theories. *Computers & Education*, 63:267 – 284.

Joachims, T. (1996). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, DTIC Document.

Johnson, D. E., Oles, F. J., Zhang, T., and Goetz, T. (2002). A decision-tree-based symbolic rule induction system for text categorization. *IBM Systems Journal*, 41(3):428–437.

Kaczmarczyk, L. C. (2013). Moo cs! *ACM Inroads*, 4(1):19–20.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kamel, M., Aussenac-Gilles, N., Buscaldi, D., and Comparot, C. (2013). A semi-automatic approach for building ontologies from acollection of structured web documents. In *Proceedings of the seventh international conference on Knowledge capture*, pages 139–140. ACM.

Kanuka, H. and Garrison, D. (2004). Cognitive presence in online learning. *Journal of Computing in Higher Education*, 15(2):21–39.

Kasimati, A. and Zamani, E. (2011). Education and learning in the semantic web. In *15th Panhellenic Conference on Informatics*, pages 338–344.

Katz, B., Borchardt, G. C., and Felshin, S. (1993). Natural language annotations for question answering. In *roceedings of the 19th International FLAIRS Conference (FLAIRS 2006)*, Melbourne Beach, FL.

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A. J., and Temelkuran, B. (2002). Omnibase: Uniform access to heterogeneous data for question answering. In *International Conference on Application of Natural Language to Information Systems*, pages 230–234. Springer.

Kazi, H., Haddawy, P., and Suebnukarn, S. (2010). *Intelligent Tutoring Systems: 10th International Conference, ITS 2010, Pittsburgh, PA, USA, June 14-18, 2010, Proceedings, Part I*, chapter Leveraging a Domain Ontology to

Increase the Quality of Feedback in an Intelligent Tutoring System, pages 75–84. Springer Berlin Heidelberg, Berlin, Heidelberg.

Konnikova, M. (2014). Will moocs be flukes. *The New Yorker*, 7.

Kop, R., Fournier, H., and Mak, J. (2011). A pedagogy of abundance or a pedagogy to support human beings? participant support on massive open online courses. *The International Review of Research in Open and Distance Learning*, 12(7).

Landauer, T. K., Mcnamara, D. S., Dennis, S., and Kintsch, W., editors (2007). *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates.

Laurillard, D. (2014). Five myths about moocs. *The Times Higher Education. Jan 16th*.

Li, J. and Wang, X. (2013). To discover and integrate the education resources based on semantic web. In *5th International Conference on Measuring Technology and Mechatronics Automation*, pages 1264–1267.

Li, Y., Bontcheva, K., and Cunningham, H. (2005). Using uneven margins svm and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, pages 72–79, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li, Y. H. and Jain, A. K. (1998). Classification of text documents. *The Computer Journal*, 41(8):537–546.

Lintean, M. and Rus, V. (2012). Measuring semantic similarity in short texts through greedy pairing and word semantics. In *Twenty-Fifth International FLAIRS Conference*.

Litherland, K., Carmichael, P., and Martínez-García, A. (2013). Ontology-based e-assessment for accounting: Outcomes of a pilot study and future prospects. *Journal of Accounting Education*, 31(2):162 – 176.

Liu, B. (2007). *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Verlag.

Liu, Z., Yu, W., Chen, W., Wang, S., and Wu, F. (2010). Short text feature selection for micro-blog mining. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1–4. IEEE.

Luo, C., Li, Y., and Chung, S. M. (2009). Text document clustering based on neighbors. *Data Knowledge Engineering*, 68(11):1271 – 1288. <ce:title>Including Special Section: Conference on Privacy in Statistical Databases (PSD 2008) – Six selected and extended papers on Database Privacy</ce:title>.

Mahajan, A. and Sharmistha, S. R. (2015). Feature selection for short text classification using wavelet packet transform. *CoNLL 2015*, page 321.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Maynard, D., Li, Y., and Peters, W. (2008). Nlp techniques for term extraction and ontology population. In *Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, pages 107–127, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Mazoue, J. G. (2013). The mooc model: Challenging traditional education. In *ELI 2013 Online Spring Focus Session 2013: Learning and the MOOC*. EduCause.

Mittal, A., Gupta, S., Kumar, P., and Kashyap, S. (2005). A fully automatic question-answering system for intelligent search in e-learning documents. *International Journal on E-Learning*, 4(1):149–166.

Mollá, D. and Vicedo, J. L. (2007). Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61.

Muoz-Merino, P. J., Pardo, A., Scheffel, M., Niemann, K., Wolpers, M., Leony, D., and Kloos, C. D. (2011). An ontological framework for adaptive feedback to support students while programming. In *International Semantic Web Conference*.

Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

Nayak, A., Agarwal, J., Yadav, V., and Pasha, S. (2009). Enterprise architecture for semantic web mining in education. In *Second International Conference on Computer and Electrical Engineering*, volume 2, pages 23–26.

Ng, H. T., Teo, L. H., and Kwan, J. L. P. (2000). A machine learning approach to answering questions for reading comprehension tests. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 124–132. Association for Computational Linguistics.

O'Callaghan, L., Mishra, N., Meyerson, A., Guha, S., and Motwani, R. (2002). Streaming-data algorithms for high-quality clustering. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 685 –694.

Onah, D. F., Sinclair, J., and Boyatt, R. (2014a). Dropout rates of massive open online courses: behavioural patterns. *EDULEARN14 Proceedings*, pages 5825–5834.

Onah, D. F., Sinclair, J., and Boyatt, R. (2014b). Exploring the use of mooc discussion forums. In *Proceedings of London International Conference on Education*, pages 1–4. LICE.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Parapar, J. and Barreiro, A. (2008). Winnowing-based text clustering. In Shanahan, J. G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D. A., Kolcz, A., Choi, K.-S., and Chowdhury, A., editors, *CIKM*, pages 1353–1354. ACM.

Parekh, V. and Gwo, J.-P. J. (2004). Mining Domain Specific Texts and Glossaries to Evaluate and Enrich Domain Ontologies. In *International Conference of Information and Knowledge Engineering*, Las Vegas, NV. The International MultiConference in Computer Science and Computer Engineering.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106.

Ramesh, A., Goldwasser, D., Huang, B., Daume III, H., and Getoor, L. (2014). Understanding mooc discussion forums using seeded lda. In *9th ACL Workshop on Innovative Use of NLP for Building Educational Applications*. ACL.

Razmerita, L., Angehrn, A., and Maedche, A. (2003). Ontology-based user modeling for knowledge management systems. In *Proceedings of the 9th International Conference on User Modeling*, UM'03, pages 213–217, Berlin, Heidelberg. Springer-Verlag.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document.

Ruiz, M. E. and Srinivasan, P. (1998). Automatic text categorization using neural networks. In *Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research*, pages 59–72.

Rus, V. and Lintean, M. (2012). A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rus, V., Lintean, M., Banjade, R., Niraula, N., and Stefanescu, D. (2013). Semilar: The semantic similarity toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 163–168, Sofia, Bulgaria. Association for Computational Linguistics.

Schleimer, S. (2003). Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data 2003*, pages 76–85. ACM Press.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

Shatnawi, S., Gaber, M. M., and Cocea, M. (2014). Automatic content related feedback for moocs based on course domain ontology. In *IDEAL*, Lecture Notes in Computer Science. Springer.

Song, G., Ye, Y., Du, X., Huang, X., and Bie, S. (2014). Short text classification: A survey. *Journal of Multimedia*, 9(5):635–643.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. Technical Report 00-034, University of Minnesota.

Studer, R. and Staab, S. (2009). *Handbook on Ontologies*. International Handbooks on Information Systems. Springer.

Stump, G. S., DeBoer, J., Whittinghill, J., and Breslow, L. (2013). Development of a framework to classify mooc discussion forum posts: Methodology and challenges. In *NIPS Workshop on Data Driven Education*.

Sure, Y., Staab, S., and Studer, R. (2006). Ontology engineering methodologies. In *In Semantic Web Technologies: Trends and Research in Ontology-based Systems*. Wiley, UK.

Tam, Y.-C. and Schultz, T. (2008). Correlated bigram lsa for unsupervised language model adaptation. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *NIPS*, pages 1633–1640. Curran Associates, Inc.

University, C. M. (2013). Open learning initiative @Online.

Valencia-García, R., Castellanos Nieves, D., Vivancos Vicente, P., Fernández Breis, J., Martínez-Béjar, R., and García Sánchez, F. (2004). An approach for ontology building from text supported by nlp techniques. In Conejo, R., Urretavizcaya, M., and Pérez-de-la Cruz, J.-L., editors, *Current Topics in Artificial Intelligence*, volume 3040 of *Lecture Notes in Computer Science*, pages 126–135. Springer Berlin Heidelberg.

Vapnik, V. N. and Kotz, S. (1982). *Estimation of dependences based on empirical data*, volume 40. Springer-Verlag New York.

Vardi, M. Y. (2012). Will moocs destroy academia? *Commun. ACM*, 55(11):5–5.

Vargas-Vera, M. and Motta, E. (2004). *MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004. Proceedings*, chapter AQUA – Ontology-Based Question Answering System, pages 468–477. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wallace, M. (2007). *Jawbone Java WordNet API*.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 977–984, New York, NY, USA. ACM.

Wang, B.-k., Huang, Y.-f., Yang, W.-x., and Li, X. (2012). Short text classification based on strong feature thesaurus. *Journal of Zhejiang University SCIENCE C*, 13(9):649–659.

Wang, T., Li, Y., Bontcheva, K., Cunningham, H., and Wang, J. (2006). Automatic extraction of hierarchical relations from text. In Sure, Y. and Domingue, J., editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 215–229. Springer Berlin Heidelberg.

Wen, D., Cuzzola, J., Brown, L., et al. (2012a). Instructor-aided asynchronous question answering system for online education and distance learning. *The International Review of Research in Open and Distributed Learning*, 13(5):102–125.

Wen, D., Cuzzola, J., Brown, L., and Kinshuk (2012b). Instructor-aided asynchronous question answering system for online education and distance learning. *The International Review of Research in Open and Distributed Learning*, 13(5).

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Xu, J., Jia, K., and Fu, J. (2008a). Research of automatic question answering system in network teaching. In *The 9th International Conference for Young Computer Scientists*, pages 2556–2560.

Xu, Y., Wang, B., Li, J., and Jing, H. (2008b). An extended document frequency metric for feature selection in text categorization. In *Proceedings of the 4th Asia Information Retrieval Conference on Information Retrieval Technology*, AIRS'08, pages 71–82, Berlin, Heidelberg. Springer-Verlag.

Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., and Ishizuka, M. (2009). Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1021–1029, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yang, S. J., Chen, I. Y.-L., Shao, N. W., et al. (2004). Ontology enabled annotation and knowledge management for collaborative learning in virtual learning community. *Educational Technology & Society*, 7(4):70–81.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420.

Yin, C., Xiang, J., Zhang, H., Wang, J., Yin, Z., and Kim, J.-U. (2015). A new svm method for short text classification based on semi-supervised learning. In *2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS)*, pages 100–103. IEEE.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12:372–390.

Zhai, C. (2008). Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies*, 1(1):1–141.

Zhang, M. and Wang, W. (2012). Research on ontology instance learning based on maximum entropy model. In *Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on*, pages 45–48.

Zhang, Z. and Liu, G. (2009). Study of ontology-based intelligent question answering model for online learning. In *2009 First International Conference on Information Science and Engineering*.

Zhen, Y. and Zheng-wan, Z. (2013). *The Design of Ontology-Based Intelligent Answering System Model in Network Education*, pages 383–390. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zheng, Z. (2002a). Answerbus question answering system. In *Proceedings of the second international conference on Human Language Technology Research*, pages 399–404. Morgan Kaufmann Publishers Inc.

Zheng, Z. (2002b). Developing a web-based question answering system. In *The Eleventh World Wide Conference (WWW 2002)*, pages 7–11.

Zouaq, A. and Nkambou, R. (2009). Evaluating the generation of domain ontologies in the knowledge puzzle project. *Knowledge and Data Engineering, IEEE Transactions on*, 21(11):1559–1572.

Zouaq, A., Nkambou, R., and Frasson, C. (2007). Building domain ontologies from text for educational purposes. In Duval, E., Klamma, R., and Wolpers, M., editors, *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 393–407. Springer Berlin Heidelberg.