Educational Data Classification Framework for Community Pedagogical Content Management using Data Mining

Husnain Mushtaq[1], Imran Siddique[2], Dr. Babur Hayat Malik[3], Muhammad Ahmed[4], Umair Muneer Butt[5], Rana M. Tahir Ghafoor[6], Hafiz Zubair[7], Umer Farooq[8]

Department of CS & IT, The University of Lahore, Gujrat, Pakistan

***Abstract*—Recent years witness the significant surge in awareness and exploitation of social media especially community Question and Answer (Q&A) websites by academicians and professionals. These sites are, large repositories of vast data, pawing ways to new avenues for research through applications of data mining and data analysis by investigation of trending topics and the topics of most attention of users. Educational Data Mining (EDM) techniques can be used to unveil potential of Community Q&A websites. Conventional Educational Data Mining approaches are concerned with generation of data through systematic ways and mined it for knowledge discovery to improve educational processes. This paper gives a novel idea to explore already generated data through millions of users having variety of expertise in their particular domains across a common platform like StackOverFlow (SO), a community Q&A website where users post questions and receive answers about particular problems. This study presents an EDM framework to classify community data into Software Engineering subjects. The framework classifies the SO posts according to the academic courses along with their best solutions to accommodate learners. Moreover, it gives teachers, instructors, educators and other EDM stakeholders an insight to pay more attention and focus on commonly occurring subject related problems and to design and manage of their courses delivery and teaching accordingly. The data mining framework performs preprocessing of data using NLP techniques and apply machine learning algorithms to classify data. Amongst all, SVM gives better performs with 72.06% accuracy. Evaluation measures like precision, recall and F-1 score also used to evaluate the best performing classifier.**

Keywords—Text mining; educational data mining; social learning; course design and delivery; technology supported learning; crowdsourced educational data mining

I. Introduction

Continuously mounting volume of data across social media and community websites has become a rich and highly potential knowledge source for a large group of users. Now a days, if people need to acquire knowledge about new subject or want to solve some particular problem, they look towards fastest, reliable and to the point information that address their needs [1][2][3]. People very often tend to utilize pedagogical values of social media like web communities, online platforms, Community Questioning Answering (CQA), generally known as questioning answering websites, crawling large amount of data from a large array of geographically distributed users with variety of expertise. Recent years witness the popularity and emergence of these Q&A websites among educational students and industrial learners who seek help and solutions of their problem with their course work and assignments [4]. Pearson's latest online annual report on social media for teaching and learning reveals that there has been an acute rise in social media in higher education institutions in recent years. It is clearly indicated from popularity community websites and social media technologies like Facebook, Twitter, StackExchange, Quora, etc [5].

Current studies reveal that a debate among educators, academicians and instructors is being done to utilize the pedagogical potential and helping end of Q&A sites to increase the productivity learning management systems. Students must be encouraged to avail such type of help available on these forums [4]. Ultimate goal of Educational Data Mining (EDM) is to facilitate the learning of students' models in an active manner to equip them with skills for which their study programs are designed for. Other than educators' debate over community Q&A sites data mining, studies say that there is potential opportunity for teachers and students to learn about variety of teaching approaches and learning behaviors respectively. Q&A sites are large repositories of vast data that can potentially be mined by pedagogical stakeholders to gain detailed insight about needs of learners and challenges [6]. Research in interdisciplinary fields of education and data sciences resulted in emergence of new field of research, Educational Data Mining (EDM). According to international EDM society, "educational data mining is an emerging discipline, concerned with the developing methods for exploring unique types of data that comes from educational settings and using those methods better understanding students, and the settings which they learn in" [5].

EDM research in computer science literature, including programming and software development, is one of the best subject fields that present themselves as rich candidates to map beginners and experts' problems in educational courses. This is because either learners or trainees in field of CS or IT possess enough skills required to utilize the educational materials available on community Q&A websites and on other social media platforms [7]. Hence, CS pedagogical stakeholders are the pioneer users of social media technologies for the sole purpose of education. This is clearly reflected from the fact that StackOverflow site is relatively larger in size as compared to the other StackExcange network which covers over 100 various topics [8]. However, manual analysis of large volume and variety of information exchanged by SO users is a laborious task that is almost seems impossible or becomes impractical in case of SO information is in large variety, huge volume and high velocity [9][10]. This study believes in application of data mining and knowledge representation methods, which already have been deployed successfully in other domains, to facilitate the process of analyzing the content of community educational forums [8].

EDM research on community Q&A, especially on SO, has mainly targeted the aspects such as answer quality measurement, users' ranking according to their knowledge, user identification and profiling, success factors of community Q&A, and subject related analysis. How to utilize knowledge of crowdsourced Q&A websites by educators to improve their teaching, delivery and coverage of subjects? And finally, how improve the learning process of learners and trainees? This paper presents a framework for text mining to discover the well-defined topics and categories which have been most frequently asked about in StackOverflow [11]. This study describes an early attempt to address the problem in relation with CS, IT and software development by proposing a Community Educational Data Mining (CEDM) framework to investigate potential and benefits of SO information to computer related educational stakeholder. Initially, this research includes six subjects for SO data management or classification. This paper describes layout and structure of CEDM and various data mining methods that can be used in CEDM to discover well defined topics and their categories which have been frequently asked [12]. This study addresses the following research questions: RQ1. How can data mining of community question answering (StackOverfkow) be exploited to provide an insight to understand CS, IT and SE related problems faced by learners? RQ2. What are best possible NLP techniques of data preprocessing to find most informative feature for each subject? RQ3. Which is the best algorithm to classify such data into respective subject with best accuracy rate?

II. Literature Review

The EDM process converts raw data coming from educational systems into useful information that could potentially have a greater impact on educational research and practice" [1]. Traditionally, researchers applied DM methods like clustering, classification, association rule mining, and text mining to educational context [11]. A survey conducted in 2007, provided a comprehensive resource of papers published between 1995 and 2005 on EDM by Romero & Ventura [12]. This survey covers the application of DM from traditional educational institutions to web-based learning management system and intelligently adaptive educational hypermedia systems.

In another prominent EDM survey by Peña-Ayala [13], about 240 EDM sample works published between 2010 and 2013 were analyzed. One of the key findings of this survey was that most of the EDM research works focused on three kinds of educational systems, namely, educational tasks, methods, and algorithms. Application of DM techniques to study on-line courses was suggested by Zaıane & Luo [14]. They proposed a non-parametric clustering technique to mine offline web activity data of learners. Application of association rules and clustering to support collaborative filtering for the development of more sensitive and effective e-learning systems was studied by Zaıane [15]. The researchers Baker, Corbett & Wagner [16] conducted a case study and used prediction methods in scientific study to game the interactive learning environment by exploiting the properties of the system rather than learning the system. Similarly, Brusilovsky & Peylo [17] provided tools that can be used to support EDM. In their study Beck & Woolf [18] showed how EDM prediction methods can be used to develop student models. It must be noted that student modeling is an emerging research discipline in the field of EDM [6]. While another group of researchers, Garcia at al [19] devised a toolkit that operates within the course management systems and is able to provide extracted mined information to non-expert users. DM techniques have been used to create dynamic learning exercises based on students' progress through English language instruction course by Wang & Liao [20]. Although most of the e-learning systems utilized by educational institutions are used to post or access course materials, they do not provide educators with necessary tools that could thoroughly track and evaluate all the activities performed by their learners to evaluate the effectiveness of the course and learning process [21].

III. Methodology

This section discusses the proposed methodology in detail which encompasses multiple stages that have been keenly observed through the literature review: This research aims to develop an software engineering knowledge classification system based on academics subject Project Management, Database Management, Software Design and Architecture, Web Development, Software Testing and Design Patterns).

Target of this research is social platform of professionals, Stack Overflow, to acquire data set where versatile people raise variety of Software Engineering questions and answer each other's questions. Manual data annotation process is performed and the annotated posts are evaluated by expert academicians and professionals. Next to annotation, data formatting and preprocessing is carried out using NLP. Supervised machine learning algorithms are used to classify data into respected classes. Moreover, proposed system is not confine to classify Stack Overflow posts, rather it is able to classify any kind of software engineering data into above given subjects. Complete process or methodology is explained in "Fig. 1".

A. Data Collection

Data collection is the first step involved in Software Engineering (SE) data classification which is done by extracting data form Stack Exchange Data dump through applying query using Stack Exchange online interface that requires reasonable and professional Structured Query Language knowledge [20]. Stack Overflow contains large quantity of software engineering knowledge and it can be utilized for educational data management. Data set of SE posts which ranges across period from 2008 to 2017 contains almost one million records. But as per research requirement only 2000 total and 500 posts of each activity were included in the experiment.

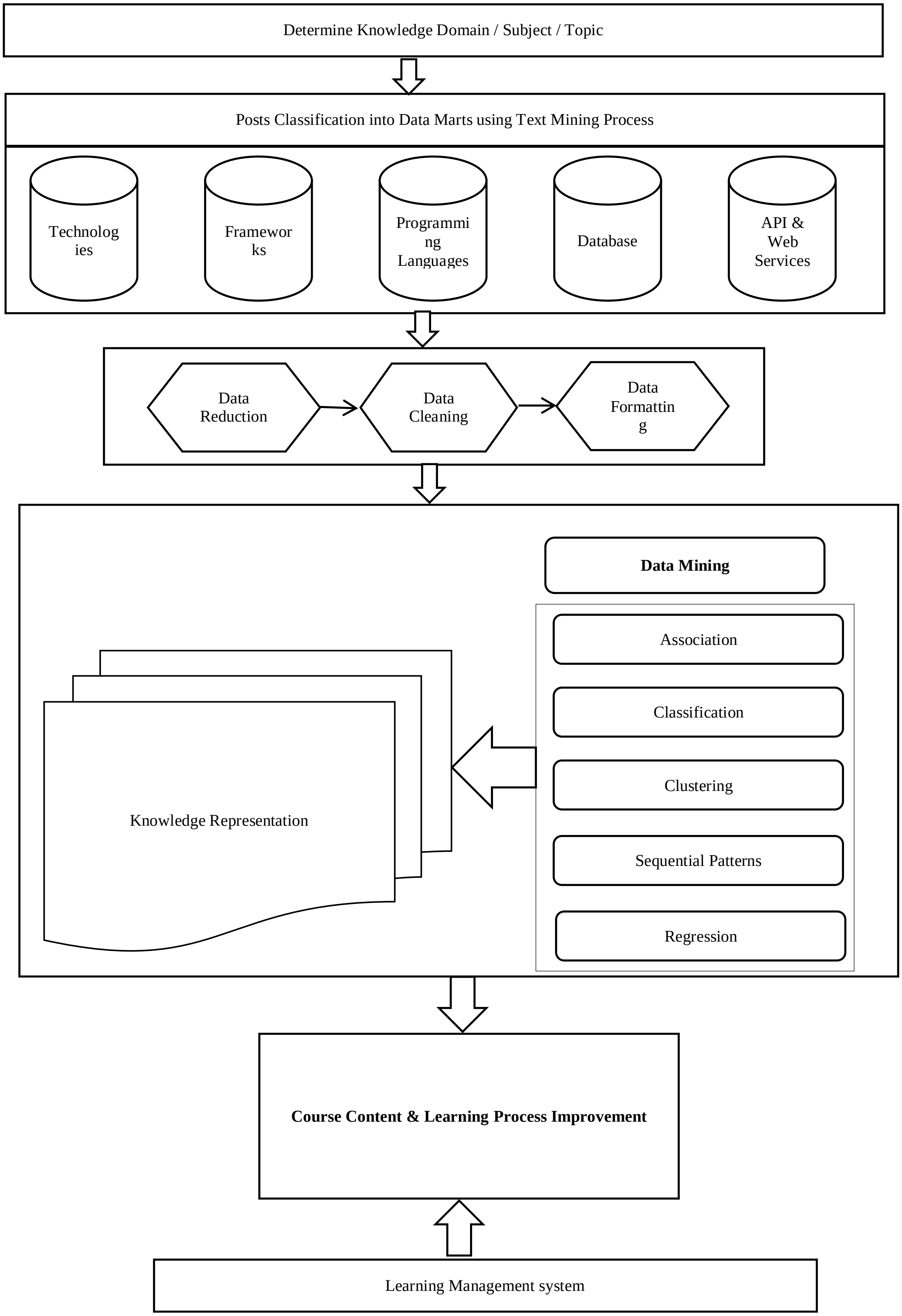

Fig. 1. Crowdsourced Educational Data Mining Framework.

B. Manual Annotation Process

To visualize the SE data and better understanding of data set, it is categorized into 6 major categories and each category contain associated subject related knowledge. Every post in selected data set is manually annotated and verified by experts.

C. Attributes Associated with Categories

Following detail reveals the pure manual categorization of posts to designate SE subject through attribute associated with each software engineering subject. Distinct attributes which differentiate each SE subjects from each other and help in manual annotation process are given as:

1) Project management: System requirement, stockholders, functional requirements (system operations), non-functional requirements (reliability, operability, performance efficiency, security, compatibility, development time, maintainability and transferability), business goals, technology contexts, performance tradeoffs, financial concerns and competitive scenarios. It also includes system refactoring and domain analysis.

2) Software design and analysis: Synthesis class attributes are design patterns, design options, metaphors, ontologies, architectural styles, software design tactics, design rationale, previous design decisions, quality attributes, high level and low level design choices, modeling standards, design improvement strategies and existing system to be integrated and future compatibility issues.

3) Design patterns: Architectural auditors, software design evaluation standards and procedures, modeling tools evaluations, design comparison techniques.

*4) Web developm*ent: Introduction to java, object oriented programming, classes, inheritance, polymorphism, collections, exceptions, streams, abstract classes and interfaces, graphical user interface, event handling, database connectivity, meta data graphics, applets, socket programming, serialization, multithreading, web application development, servlet, java server pages, java beans, model view controller, layers and tiers, java server pages standard tag library, java server faces, web services.

5) Database management systems: Basic database concepts, database architecture, database planning, conceptual database design, logical database design, transforming e-r design to relational design, data definition languages, data manipulation languages, normalization and demoralization, physical database design, database tools, structured query language (SQL), data storage concepts, indexes and views, transaction management, concurrency control.

6) Software quality assurance: Software quality, software defects, reasons of poor quality, quality laggards, project management approaches, cost and economics of SQA, quality measurements, software requirements and SQA, quality attributes of requirements document, software design model and software design defects, quality design concepts, programming and SQA, SQA reviews, software inspections, software testing **-** WBT techniques, BBT techniques, testing strategies, debugging, test planning, automated software testing, test cases, introduction to quality metrics, a process model of software quality assurance.

D. Software Engineering Posts Pre Processing

Post preprocessing is most vital step in software engineering subjects classification process. As the acquired data set is taken from social platform where versatile people with respect to software engineering knowledge share their queries, problems or answers of other’s questions in their own style and mostly posts contain variety of data like Hash tags, HTML tags, coding scripts, short texts, programming outputs etc. Almost each post contains both, useful and some useless data so it is needed to be clean it [16].

Firstly, text is made free form all irrelevant and noisy data which comprises of semicolons, quotes, question marks, exclamation marks, notations, tags, code, process results etc. Pre-processing of architectural posts data set include following steps.

1) Tokenization: Tokenization refers to a technique in which tokens (words in textual data) are extracted from a textual document by splitting sentences of textual document into tokens by delimiter [17]. A textual document consists of many words arranged in sentences. These words are separated by some delimiters in sentences like full stop, comma, hyphen, space etc. Firstly in pre-processing, tokenization is done of each textual document in the dataset. In it tokens are generated by breaking long sentences in small tokens separated by space delimiter [18].

2) Stop-word removal: All words in a textual document are not equally important in conveying context of text. Irrelevant and less informative words should be removed from dataset or corpus for effective performance of machine learning algorithms while performing classification tasks [22].

In this step text in tokenized documents is cleaned from all useless and meaningless words. A stop-word describes a word with little meaning (Scott & Matwin, 1998). Example of these words are ‘The’, ‘is’, ‘also’, etc.

3) Removal of programming content: Software engineering process contains software development process as sub activity so often people post programming code in their posts in order to make their post most elaborative and explanatory for its easy understanding. Programming contents are not part of dictionary and programming language syntax contains no standard words recognized by wordNet [23]. So before performing stemming, lemmatization and auto spell correction, it is also needed to remove programming contents form posts data.

4) Removal of tags: Removal of HTML tags is also part of data preprocessing as the data belongs to software engineering domain and best features achieved after proper removal of unwanted portion form training data set.

5) Spell checking and correction: Community discussions or review text data sets contains frequent words or phrases which are not part of standard lexical dictionaries and not recognized by search engine optimization algorithms and other machine learning models. Regular expressions and manually prepared data dictionaries are used to fix such types of noise [21].

6) Stemming: Every word in text comes from its root word but cannot be same in text. As an example, there is little difference in meaning of two words; 'Hate' and 'Hates' [24]. So to solve this type of issue in text classification and information retrieval solution, a technique is adopted which is called stemming. Stemming in an approach which is used mostly in linguistic and information retrieval to reduce words to their base stem or root word. For example in English language stemming, an stemming algorithm which is called stemmer convert words 'liked', 'likely', 'likelihood', 'liking' to base word 'Like'. technique(Collection, 2017) to converts all tokens to their base stems Table I.

TABLE I. EXAMPLE APPLICATION OF PREPROCESSING STEPS

Stack Over Flow Post	StackOverflow Post After Stemming
Migrate to a New Designed System	Migrat new design system
Importing associations, dependencies etc. from PHP code in Enterprise architect	Import assoc depend php cod enterpris architect

E. Feature Extraction

In this phase, different techniques were applied to extract useful and most informative features. Study incorporates Bag of Words, TFIDF and N-grams (1–4) for feature selection.

Selecting right features from all features is tough job but it improves overall performance of system [20]. Following are the techniques which are applied in the data set.

1) TFIDF: TF (Term Frequency) indicates the number of times a specific term appears in a document as shown in Equation 1 [22]. IDF (Inverse Document Frequency) is a numerical weight which is used to measure the importance of a specific term in collection of text document [17]. IDF reduces the weight of those terms which appear frequently in a text document and increase the weight of rarely occurred terms. In feature extraction, TF-IDF is a statistical technique to find out importance of words in corpus.study follows the approach to compute statistic for each feature (uni-gram, bi-gram, tri-gram, quad-gram) of each document (post) related to each class (developer) [19]. Then in this way, pre-processed dataset is converted in document vector form which represents each post data.

$$TF = \frac{\text{frequency of word}}{\text{total no.of words in document}} \quad (1)$$

One other way to compute term frequency is logarithmically scaled value. Let t denotes particular term in the document and d represents document in the corpus, then following formula calculates TF statistical value of each term in a document.

$$TF(t,d) = 1 + \log(tf_{base}(t,d)) \quad (2)$$

Here tf_{base} is the function which computes and returns frequency of term t in the document d as shown in Equation 2.

Inverse document frequency is the value which tells how a term is unique and rare across number of documents of a particular class as shown in Equation 3.

$$idf_{(t,D)} = \log \frac{\mathrm{N}}{|[\mathrm{d} \in \mathrm{D} : \mathrm{t} \in \mathrm{d}]|} \quad (3)$$

Then finally TF-IDF in Equation 4, value of each term is calculated by simply multiply scores of tf and idf [18].

$$TF - IDF_{(t,d,D)} = tf_{(t,d)} \times idf_{(t,D)} \quad (4)$$

At the end of this step, term to document vector form of each document across all classes which contain terms with their tf-idf scores.

2) N-Gram (unigram, bigram, trigram, quad-gram): In textual data containing n items, n-gram is a linked sequence of items from that text sample. In a text these items may refer as letters, syllables, words, base pairs etc. The source of n-grams are commonly text or speech pattern [14]. Sequence of n words from any given text is referred as n-gram as shown in "Fig. 2". Bi-gram describes words that are two words sequence pattern from given text, similarly tri-gram is three words sequence and quad-gram is four words sequence pattern. For example here is a piece of text "this is architecture post". Uni-grams of this text are 'this', 'is', 'architecture. 'post. Bi-grams generated from this text are 'this is', 'is architecture, 'architecture post as clear in Table II.

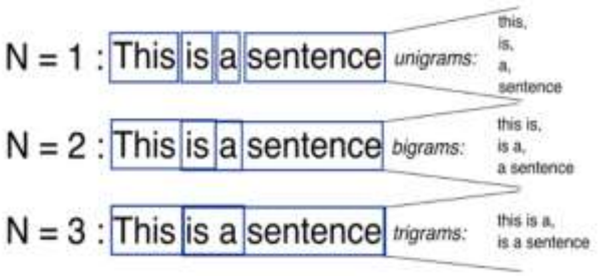

Fig. 2. Depicts the N-Gram Tokenization.

TABLE II. EXAMPLE OF CONVERTING TOKENS TO BI-GRAM, TRI-GRAM AND QUAD-GRAM

N-gram	Tokens
Uni-grams	Importing associations, dependencies etc. from PHP code in Enterprise architect
Bi-grams	Importing associations, dependencies etc, from PHP, code in, Enterprise architect
Tri-grams	Importing associations dependencies, etc from PHP, code in Enterprise, in Enterprise architect
Quad-grams	Importing associations dependencies etc, associations dependencies etc from, dependencies etc, associations dependencies, associations dependencies from PHP

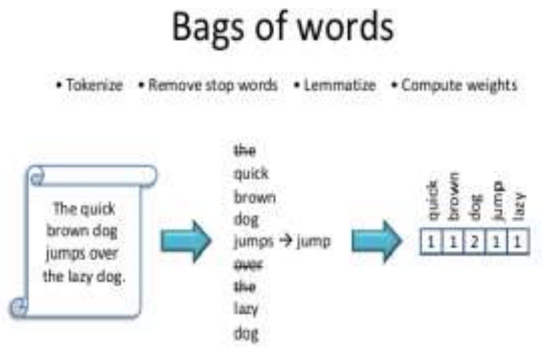
Bags of words
• Tokenize • Remove stop words • Lemmatize • Compute weights
The quick brown dog jumps over the lazy dog.
~~the~~
quick
brown
dog
jumps → jump
~~over~~
~~the~~
lazy
dog
quick
brown
dog
jump
lazy
1 1 2 1 1

Fig. 3. A Sentence Converted to Bag of Words Features.

3) Bag of words model: The bag-of-words (BOW) model considers each post as a set of words that each occurs a certain number of times. The representation of the document is entirely order less, as each word is treated independently of the previous and upcoming word. As an example, there is a data set consisting of only two messages: The cat is better than the dog and: The weather is better than yesterday.

Vector one 2 1 1 1 1 1 0 0

Vector two 1 0 1 1 1 0 1 1

As the number of samples grow, the number of unique words will increase as shown in "Fig. 3". Since each unique word is represented by a specific position in the vector, these vectors will naturally grow larger as well. The vector will have the length of the total number of unique words that exists in the data set.

F. Classifiers used in Assessment Process

1) Naïve bayes: It is a classifier which does probabilistic classification which is completely depend on features. For each feature, individual classification is done. It formulates labels of class and then within those classes text probability is calculated. In literature, Naïve Bayes is tremendously user for text classification. In classification matters, Naïve Bayes stands at better position and literature professed it better than others. It works well in text and numeric data and it very easy to implement. Correlated features effect performance of Naïve Bayes algorithm.

2) Support vector machine: Support vector machine classifier model is efficient for text classification task. It working based on multidimensional hyper planes which are made to make separation between different classes or labels. It is based on classification algorithm proposed by Boser ,Guyon and Vapnik in 1992(Boser, Laboratories, Guyon, Laboratories, & Vapnik, n.d.,1992). In text classification, number of features to be deal is very large in form or terms or words, so SVM can be used as it can easily deal with large amount of features. In this study features are terms or words from posts and architectural process activities ate classes. Svm can be efficiently used to classify features in multi-dimensional hyper planes which separate features to the boundary of their particular class.

3) K Nearest neighbor: It is simplest algorithm used in machine learning. It is lazy and instance base learning. It is mainly used in classification and regression analysis. In this approach K similar documents a considered. To make a verdict about existence of a post in anticipated class, it computes the similarity of all the documents that exists in the training set. The class with highest probability in the neighbored is assigned to specific defined category is very effective but vital cons of this algorithm are high computational time and discover ideal value of K is problematic.

IV. Experimental Evaluation

In this section, evaluation of the proposed software engineering subjects posts classification system is carried out. As this is an early approach to classify such data under educational context in multiple activities using natural language processing and machine learning so the study does not have described any benchmark against which performance of the proposed system is to measure. Results of different classifiers are compared.

A. Data Acquisition

Date was acquired from Stack Exchange Data dump using structured query language which resulted about one million software architectural posts. From overall data, only 2000 posts data and divide them into 4 architectural activities i.e. Analysis, Synthesis, Evaluation and Implementation. The whole dataset records are divided into two parts. One is used for training which contains 1400 records and other part is used for testing which contains 600 records. Python editor was used for the experimental work. The brief overview of the dataset is outlined below in table.

B. Evaluation Measures

To evaluate the subjects classification, standard evaluation methods used in previous text classification studies i.e. accuracy, precision, recall and F-measure. Every classifier result is presented in a table form to distinguish the correct predictions from the incorrect ones for each class. This table is called as confusion matrix. In this matrix:

TP = Number of posts correctly assigned to each class.

FP = Number of posts incorrectly assigned to each class.

FN = Number of posts incorrectly rejected to each class.

TN = Number of posts correctly rejected to each class.

C. Experimental Evaluation of Proposed Classification Approach

In this section proposed classification technique is evaluated by conducting three experiments with the data set. Each experiment is conducted to measure the effectiveness and overall performance of the entire classification system. Experiment 1 classify software architecture posts using NLP rule, BOW (Bag of Words) approach and gives initial but satisfied classification results. Experiment 2 is performed using TFIDF approach by applying different threshold values. Experiment 3 follows the N-Gram approach which consumes unigram, bigram, trigram and quad-gram techniques by combining with TFIDF and gives better results only in case of uni-grams while other N-grams lack in result. All above mentioned experiments are applied after applying all preprocessing steps on data set.

D. Experimental Setup

The IDE and experimental setup for experiment will remain same for each experiment. The experiment conducted on the data set contains 1400 instances for training data and 600 instances for text data. All these instances belong to 6 different classes i.e. Requirement Engineering, Architecture and Design, Software Implementation and Software Testing. Each class contains 350 instances for training set and each of four classes in test set contain 150 instances. Hence total 2000 instances, 500 for each class are utilized in experiment.

The 70% samples from dataset are filtered using the pre-processing techniques described in chapter 3. The pre-processing methods tokenization, stop words removal, spell checker and word completion, stemming are applied to each sample of train data. World list is updated on each step of pre-processing.

1) Experiment 1: The main objective of this experiment is to illustrate the classification accuracy of proposed approach using Bag of Word (BOW). The experiment has been carried out on dataset which is used to classify the software architectural posts. Here, experimental setting are changed and instead of using the aforementioned experimental setup, new experimental step to classify the status using character N-gram is created. Each class samples are broken into tokens through which N-gram lists is generated. To classify the software engineering posts into their respective class then map of all N-gram list is calculated which is further used to compute the hash map score. The following table shows the hash map score of all classes.

This indicates how many SE posts are classified correctly and how many are classified are into wrong classes. By experimental evaluation, it is observed that the better accuracy of proposed classification approach is achieved i.e. 69.12% as shown by “Fig.4” and in table 3.

2) Experiment 2: First Experiment was conducted using bag of word approach. Second Experiment using TFIDF by apply counter_vectorizer() method of SKlearn library of machine learning toolkit. Results are shown in table 4 and “Fig. 5”.

3) Experiment 3: The objective to perform this experiment is to demonstrate the classification accuracy of the proposed NLP approach using the stop words list and TFIDF. Same dataset has been used to carry out experiment which has been used for categorization the status. Here existing stop word list is used in combination of TFIDF.

Following the preparation of N-gram pattern, TFIDF matrix is used to compute the score of each individual class which is computed by total number of words in corpus divided by their individual frequencies as shown in table . Training model is generated and each classifier used in the experiment. Following are result as shown in tabular and chart forms as in “Fig. 6”, “Fig. 7”, “Fig. 8”, “Fig. 9” and Table V.

TABLE III. POST CLASSIFICATION USING BAG OF WORDS

Features	**No. of Features**	**Naïve Bayes**	**SVN**	**KNN**
Bag of Words	3529	67.43	69.12	63.51

TABLE IV. TFIDF FEATURES RESULTS

Features	**No. of Features**	**Naïve Bayes**	**SVN**	**KNN**
TF-IDF	2368	70.61	73.86	68.42

TABLE V. RESULTS USING N-GRAM FEATURES SET

Features	**No. of Features**	**Naïve Bayes**	**SVN**	**KNN**
Unigram	2962	70.62	73.54	62.41
Bigram	13535	62.32	68.05	59.92
Trigram	11356	51.03	48.64	47.73
Quad Gram	17684	38.63	40.26	42.08

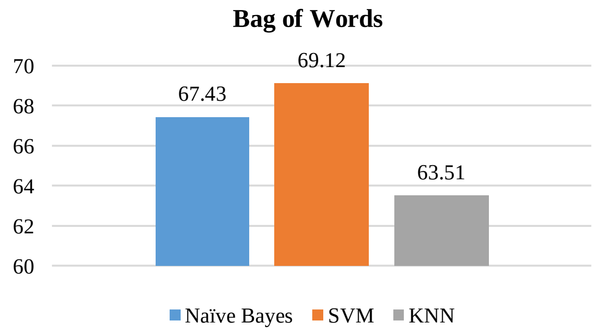

Fig. 4. Post Classification using Bag of Words.

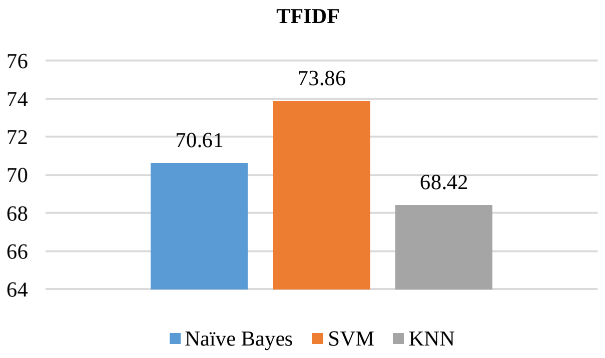

Fig. 5. TFIDF Features Results.

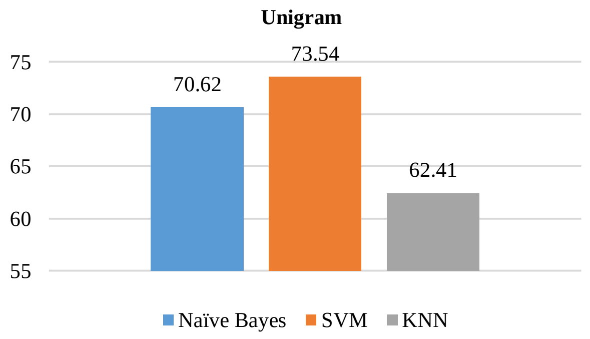

Fig. 6. TFIDF with Unigrams Tokens.

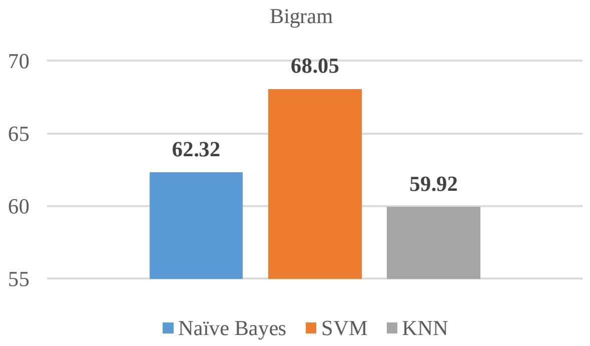

Fig. 7. TFIDF with Bigrams Tokens.

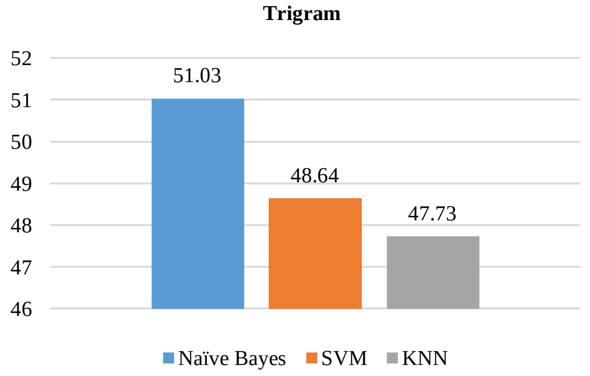

Fig. 8. TFIDF with Trigrams Tokens.

E. Comparison of Accuracies of Data Mining Techniques Followed in Experiment

Overall accuracy performance of the proposed system with chosen classifiers are depicted in figure 9. it shows the all classifiers result with respect to different N-gram patterns. Three classifiers are used on different N-gram patterns. After this experiment, SVM performs well with unigram and bigram features with 73.54% and 68.05% accuracy whereas KNN gives better results with trigram and quad gram features with 47.73% and 42.08% accuracy.

Naïve Bayes accuracy lies between SVM and KNN. Detail of all classifiers along with different n-gram pattern are articulated below in the "Fig. 10".

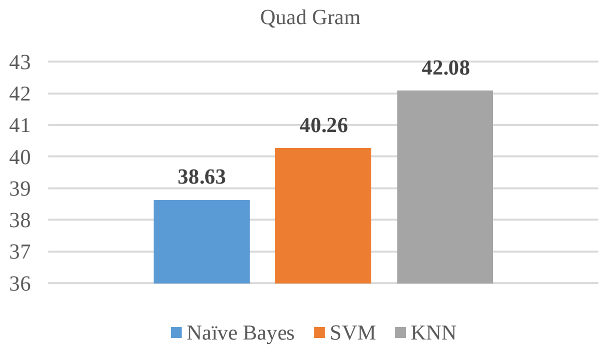

Fig. 9. TFIDF with Quad Grams Tokens.

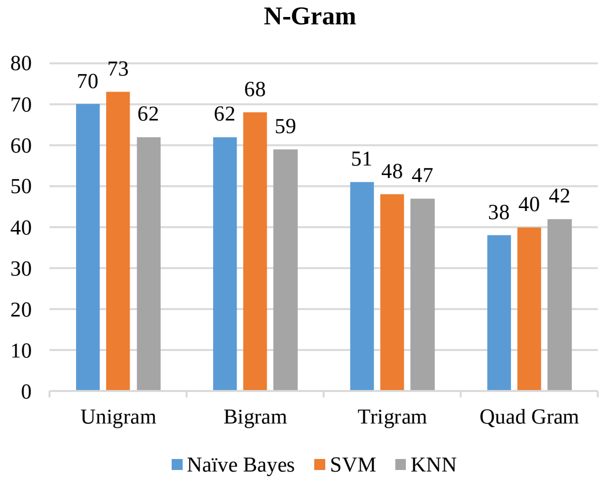

Fig. 10. Results using N-Gram Features Set.

F. Comparison of Precision and Recall Scores

Table VI gives the precision, recall and F1 score of three algorithms on different n-gram patterns which are unigram, bigram, trigram and quad gram. Three algorithms are applied one by one on the dataset and acquired results. Table VI depicts the comparison of all the features on three different classifiers.

TABLE VI. CROSS VALIDATION OF CLASSIFIERS USING TFIDF AND NGRAMS

Features	Classifier	Precision	Recall	f1-score
Unigram	Naïve Bayes	0.75	0.76	0.75
	SVM	0.82	0.65	0.74
	KNN	0.71	0.58	0.83
Bigram	Naïve Bayes	0.75	0.76	0.75
	SVM	0.82	0.65	0.74
	KNN	0.71	0.58	0.83
Trigram	Naïve Bayes	0.75	0.76	0.75
	SVM	0.82	0.65	0.74
	KNN	0.71	0.58	0.83

G. Cross Validation of Performance of Three Models

In this experiment, k-fold technique is applied on out dataset to verify results of the proposed model and divided the dataset into 10 folds (f1, f2, f3 . . . . f10) of equal size. Firstly, classifier is trained with f1 to f9 folds and tested for f10 folds then trained with f1 to f8 and f10 folds and tested for f9 and so on.

The overall comparison of performance of three classifiers is depicted on the graph below. In the graph, Support Vector Machine has edge of the slop far off from the left. So, it shows a greater performance as compared to KNN and Naïve Bayes.

H. Discussion

This research is conducted with sole objective to find a machine learning based mechanism to classify and manage pedagogical knowledge residing over crowdsourced communities. It was an initial step to make Q&A communities a part of well managed online libraries. This research was confined to only six subjects of software engineering domain which do not cover the even major areas of SE domain. Study followed supervised machine learning based experiments therefore manual annotation process was quite hectic and time taking. Moreover, preprocessing of community data, especially SE domain data which not totally natural language based, was also a novel job to perform. Results obtained and validations process depicts satisfactory performances of algorithms especially SVM. There are numerous feature extraction techniques and machine learning techniques which can be incorporated to manage educational knowledge across the online communities.

V. Conclusion

Conclusion: Goal of this study is to unveil the potential of crowdsourced experts and communities across the internet to mitigate subject related problems faced by learners through identifying frequently posted questions and map the discovered knowledge, using data mining techniques, on learning management system to improve overall learning process by enhancing course content management. Moreover, this paper distinguishes the croudsourcing approach of education data mining as more suitable and rip with knowledge as compared to EDM using learning management system data. The proposed framework determines subject or domain of knowledge to be discovered and find knowledge patterns using data mining techniques after the cleaning and formatting community Q&A data. Then relates the discovered subject related knowledge is integrated with learning management system to improve course content quality and pedagogical methods to enhance students' learning process. The study is not limited to single knowledge domain or subject, rather can be implemented to almost complete spectrum of educational studies. In our future work, we will implement this framework to find relationship patterns between crowdsourced community literatures with learning management system of different subjects through classification, clustering and association techniques. Moreover, we will extend our study to remaining core knowledge base area over SO and on other Q&A communities.

Refrences

[1] Siti Rochimah, Rizky Januar Akbar Achmad Arwan, "Source Code Retrieval on StackOverflow Using LDA," in 3rd International Conference on Information and Communication Technology (ICoICT), 2015, pp. 295-299. (A. Heß, 2008)

[2] Michael English, Abdulhussain E. Mahdi Arash Joorabchi, "Text mining stackoverflow, An insight into challenges and subject-related difficulties faced by computer science learners," Journal of Enterprise Information, vol. Vol. 29 No. 2, 2016, pp. 255-275, August 2015.

[3] GÜLFEM IŞIKLAR ALPTEKİN ASLI SARI, "An Overview of Crowdsourcing Concepts in Software Engineering," International Journal of Computers, pp. 106-114, 2017.

[4] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions," in WICSA , A. Jansen and J. Bosch, ",” in WICSA, 2005, pp. 109–120., 2005, pp. 109-120.

[5] Filippo Lanubile, Maria Concetta Marasciulo, Nicole Novielli Fabio Calefato, "Mining Successful Answers in Stack Overflow," , 2015.

[6] R. E. D. Silva, R. L. Rodrigues, J. C. S. Silva and A. S. Gomes J. L. C. Ramos, "A Comparative Study between Clustering Methods in Educational Data Mining," in IEEE LATIN AMERICA TRANSACTIONS., 2016, pp. 355-3361.

[7] Shuang Peng, Lin Wang, Bin Wu Juan Yang, "Finding Experts in Community Question Answering Based on Topic-Sensitive Link Analysis," in IEEE First International Conference on Data Science in Cyberspace, 2016, pp. 55-60.

[8] Nitya Upadhyay and Vinodini Katiyar, "A Survey on the Classification," International Journal of Computer Applications Technology and Research, pp. 725-728, 2014.

[9] P. Clements, and R. Kazman L. Bass, Software Architecture in Practice, 3rd ed., Addison-Wesley Professional, Ed., 2012.

[10] P. Clements, and R. Kazman L. Bass, Software Architecture in Practice.: Addison-Wesley Professional, 2012.

[11] Olga Baysal, David Lo, Foutse Khomh Latifa Guerrouj, "Software Analytics: Challenges and Opportunities," in IEEE/ACM 38th IEEE International Conference on Software Engineering Companion, Austin, USA, 2016, pp. 902-903.

[12] R Lakshmana Kumar, Abhijith Surendran, K Prathap M Amala Jayanthi, "Research Contemplate on Educational Data Mining," in IEEE International Conference on Advances in Computer Applications (ICACA), 2016, pp. 110-114.

[13] R Lakshmana Kumar, Abhijith Surendran, K Prathap M Amala Jayanthi, "Research Contemplate on Educational Data Mining," in IEEE International Conference on Advances in Computer Applications (ICACA), 2016, pp. 110-114.

[14] M. Riebisch, and U. Zdun M. Soliman, "Enriching architecture knowledge with technology design decisions," in WICSA, 2015.

[15] D. Pagano and W. Maalej, "How do open source communities blog," in Empirical Software Engineering, vol. 18, no. 6, 2013, pp. 1090–1124.

[16] Steve Mcconnell, Code Complete, 2nd ed.: (Microsoft Press, 2004.

[17] Emad Shihab Meiyappan Nagappan, "Future Trends in Software Engineering Research for,".

[18] Matthias Galster, Amr R. Salama, Matthias Riebisch Mohamed Soliman, "Architectural Knowledge for Technology Decisions in Developer Communities," in IEEE/IFIP Conference on Software Architecture, 2016, pp. 128-133.

[19] Muhammad Assaduzamman, Chanchal K. Roy, Kevin A. Schneider Muhammad Ahsanuzamman, "Mining Duplicate Questions in Stack Overflow," in IEEE/ACM 13th Working Conference on Mining Software Repositories, 2016, pp. 402-412.

[20] Pankaj Dhoolia, Rohan Padhye, Senthil Mani and Vibha Singhal Sinha Neelamadhav Gantayat, "The Synergy Between Voting and Acceptance of Answers on StackOverflow, or the Lack thereof," in 12th Working Conference on Mining Software Repositories, 2015, pp. 406-409.

[21] J. Koehler, F. Leymann, R. Polley, and N. Schuster, O. Zimmermann, "Managing architectural decision models with dependency relations,integrity constraints, and production rules,”," Journal of Systems and Software, vol. vol. 82, pp. 1249–1267, 2009.

[22] Patrick Hennig,Tom Bocklisch, Tom Herold, and Christoph Meinel, Hasso-Plattner Philipp Berger, "A Journey of Bounty Hunters: Analyzing the Influence of Reward Systems on StackOverflow Question Response Times," in IEEE/WIC/ACM International Conference on Web Intelligence, 2016, pp. 644-649.

[23] Gregory Piatetsky-Shapiro, Advances in Knowledge Discovery and Data Mining, Gregory Piatetsky-Shapiro, Padhraic Smyth and Ramasamy Uthurusamy Usama M. Fayyad, Ed. America: American Association for Artificial Intelligence Press, 2017.

[24] Zhangyuan Mengy, Beijun Sheny, Wei Yinz Yunxiang Xiongy, "Mining Developer Behavior Across GitHub and StackOverflow," , 2017.