



MAPPING LEARNER’S QUERY TO LEARNING OBJECTS USING TOPIC MODELING AND MACHINE LEARNING TECHNIQUES

SOUVIK SENGUPTA*, SAURABH PAL† AND PIJUSH KANTI DUTTA PRAMANIK‡

Abstract. Inquiry-based learning supports the independent knowledge development of the learner in an e-learning environment. It is crucial for the learner to obtain the appropriate Learning Object (LO) for the intended query. Mapping a learner’s query to the right LO is a challenging task, as keyword-based searching on the topics or content does not guarantee the best result for various reasons. A query that apparently connects a topic may also implicitly refer to multiple other topics. Besides, the content of an LO with the same topic name often varies over different portals. Therefore, there is always a need for a method to automatically identify the latent topics of the query and then find the most relevant LO that covers the query. This paper aims to build a recommender system that maps a given input query to a suitable LO based on the most appropriate matching of learning contents. The proposed work employs an amalgamation of different supervised and unsupervised methods of natural language processing and machine learning. The machine learning model is trained on a handcrafted dataset to map queries into predefined topics. The proposed algorithm also leverages a dynamic topic modeling technique on learning content collected from three popular e-learning portals and uses a similarity score to map the learner’s (user) query to the most appropriate LO.

Key words: Natural Language Processing, Machine Learning, Topic Modeling, LDA, Learning Object

1. Introduction. Inquiry-based learning is a method of active learning that starts by posing questions, problems, or keywords and then generating facts and information about a topic related to the query. In an e-learning environment, it helps the learner to develop knowledge on their own. Inquiry-based learning is one of the popular forms of problem-based learning (PBL), among others like simulations, case studies, guided design, and project-based learning, where the problem comes before instructions and knowledge [4] [5] [8]. E-learning provides a powerful platform for inquiry-based learning as it facilitates the enquirer to seek information from an extensive repository of learning objects (LOs). An LO is a collection of content, example, and assessment items that are combined based on a single learning objective [9] [18].

Despite the availability of web search engines and in-portal searching tools, the common problem that inquiry-based learning confronts is retrieving precise information (in terms of LO) for user queries [10]. Since all search engines mostly rely on keyword-based metadata, if queries are ill-defined or missing a vital keyword, it fetches LOs that are superfluous and often fail to make any meaningful connection to the topic of a query [11]. On the other hand, even if the topic associated with the query is known, sole topic-based searching is also insufficient, as the same LO can be described under different topics at different e-learning portals. For example, the query “How is polymorphism supported in java?” may seem to be related to the topic “polymorphism”. However, the required LO can actually be found at different portals under different topics like “inheritance”, “method overriding”, and “method overloading”. This necessitates identifying the latent topic(s) associated with the query and the LOs so that topics can be matched to identify the most appropriate LO for a query.

In this paper, we address three specific problems of query-based learning, which are:

- Finding the most appropriate LO for a query.
- Finding the latent topics related to the query and LOs.
- Matching one topic model to another.

A suitable methodology is proposed in this work to automate the process of mapping the learner’s query to the appropriate LO in an e-learning environment. For identifying latent topics from the query, we employ both supervised and unsupervised techniques, and dynamic topic modeling is applied as a method for summarizing.

*Aliah University, Kolkata, India (mesouvik@hotmail.com).

†Swami Vivekananda Institute of Science and Technology, Kolkata, India (saurabhpa12007@gmail.com).

‡Galgotias University, Greater Noida, India (pijushjld@yahoo.co.in). Corresponding author

Finally, the selection of LO for a query is based on the proximity between the topic models of LOs and the topic model of the query. The notable contributions of this work can be briefed as follows:

- A dataset containing 1000 questions on ‘object oriented programming’ labeled with topic name.
- Comparative analysis of machine learning models to predict the topic from the learner’s query.
- Topic modeling for LOs on three popular e-learning portals.
- A proposed method for similarity measure within the topic models.

The rest of the paper goes as follows: Section 2 presents a review of the related works, Section 3 illustrates the proposed methodology in detail, Section 4 presents the results and analysis, and Section 5 has the conclusion.

2. Related Work. The related works can be broadly classified (with some overlapping) into two categories - a) mapping user queries to predefined classes or topics using feature selection and supervised machine learning approaches, and b) topic modeling and topic identification using the unsupervised approach.

Many works have been carried out towards mapping users’ queries to appropriate topics using different machine learning techniques. These are not only limited to e-learning but also have broad applications in tagging community questions with predefined classes, clustering and segmenting questions, and improving search results for recommendation systems.

It is evident from the reviewed works that word embedding and feature selection play a pivotal role in topic classification with machine learning models. Yang et al.[19] proposed a topic-oriented word embedding approach for query classification. This topic-oriented word embedding with fine-tuning of the Word2Vec model has shown high precision of 95.73% on the Baidu and Sogou datasets. Gulzar et al. [6] presented a course recommender system to recognize the intent and requirements of the user’s query. The query and the course are represented using the n-gram feature set and mapped to classes with the help of domain ontology. Li et al. [7] used a hierarchical classifier to classify questions based on their semantics. The questions are mapped into a hierarchy of six coarse classes and 50 fine-grained classes. The classifier uses SNoW learning architecture, where two simple classifiers are combined and used for classification. The first classifier classifies the question into coarse classes, while the second classifier classifies the question into fine-grained classes. This work uses six feature sets, where each of the sets consists of an incremental combination of primitive feature types like words, pos-tags, chunks, head chunks, named entities, and semantically related words. This empirical study showed satisfactory results with more than 90% overall precision. In a similar work, Nguyen et al. [16] described a machine learning approach for question classification based on coarse and fine-grained question classes on the TREC dataset. This work primarily focuses on feature selection methods. Different lexical, syntactical, and semantic features are extracted. Integrating these features into a unique form allowed a better feature representation of the questions for classification. Different supervised learning algorithms such as k-nearest neighbors (KNN), naive Bayes (NB), decision tree (DT), and support vector machine (SVM) are used for the classification task. The experimental result shows better classification accuracy by introducing new features than standard features for coarse and fine-grained classes.

Many researchers have also worked on mapping topics to questions using the Community Question Answering (CQA) dataset. Qu et al. [13] illustrated the problems of question topic classification using an extensive real-world CQA dataset. This work empirically evaluates the different models for classifying questions into CQA categories. This work depicts a comparative analysis of the usefulness of n-Gram features over bag-of-word features. It also employs a variety of flat classifiers, NB, Maximum Entropy (ME), and SVM, and combines them with state-of-the-art hierarchical models for question classification. The findings show that SVM outperforms other models in terms of effectiveness, while NB takes the shortest training time. In a similar work, Singh et al. [14] proposed a methodology to classify a new question into one of the hierarchical categories of a CQA portal. A translation model is used to retrieve questions from the corpus that is lexically and semantically similar to the input question. The system employs a near neighborhood-based classifier on the retrieved similar questions to detect the class for the new question.

On the other hand, most of the works related to topic identification for a query have employed Latent Dirichlet Allocation (LDA). Bhattacharya et al. [1] described a method to classify a query into a topic class by considering the query keywords distributed over various topics. This work assumes that queries consisting of keywords may belong to multiple topics, representing overlapping concepts. In this work, LDA is applied to the entire corpus to group the documents into topics consisting of unique words forming the topic models.

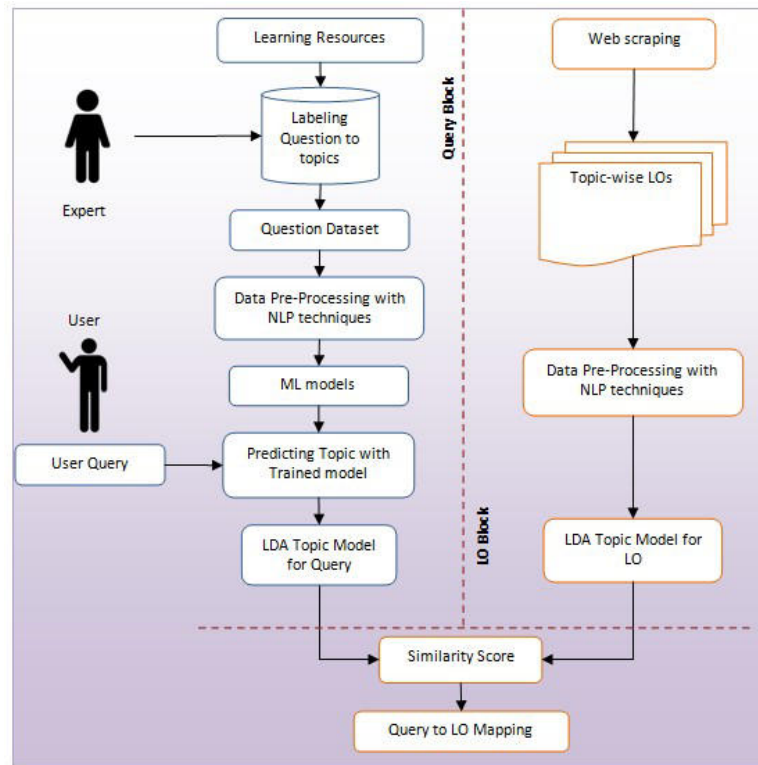


Fig. 3.1: Architecture for user query to LO mapping

Subsequently, a sparse representation-based classifier (SRC) is applied over the topic models to predict the topic class of the query. In a similar work, Zhang et al. [20] depicted an approach for capturing the word-semantic similarities between a user query and questions in the repository by introducing topic modeling. In this work, the LDA model is used to identify the questions' topics and subsequently generate the topic model. The topic model helps to map the stored Q&A pair appropriately into the topic space. An unsupervised clustering approach is used to deduce the similarity between the user's input query and the questions in the focused topic space. Subsequently, the most similar questions are obtained from the Q&A repository. In a similar work, Das et al. [3] presented an unsupervised topic modeling for auto-generated questions. It first performs a phrase mining operation on a given set of documents and checks if a set of words or terms, with a particular sequence, can occur more frequently than others and then creates bag-of-phrases. Then it employs constrained topic modeling using LDA, which uses this bag-of-phrases for the topic model. In all these works, LDA is predominantly used for topic modeling and extracting latent topics.

Most of the above-reviewed works are either focused on the classification of queries into predefined topics or on identifying latent topics dynamically from the question repositories. However, none of the works have attempted to use topic modeling and classification techniques jointly to identify the most appropriate LO for the user's query. In this work, the proposed methodology combines the best of the two worlds, i.e., a) training supervised models with necessary feature extraction on a labeled dataset and b) mapping the prediction with a dynamic topic model using LDA.

3. Methodology. Figure 3.1 depicts the framework of the proposed methodology. The working modules of the proposed framework are organized into two sets of components – the query block and the LO block. The operations of the blocks are independent of each other, but the resultant topic models are collectively assessed for similarity checks to map a query to its corresponding LO.

The query block has two components - first, it trains a machine learning model for classifying questions

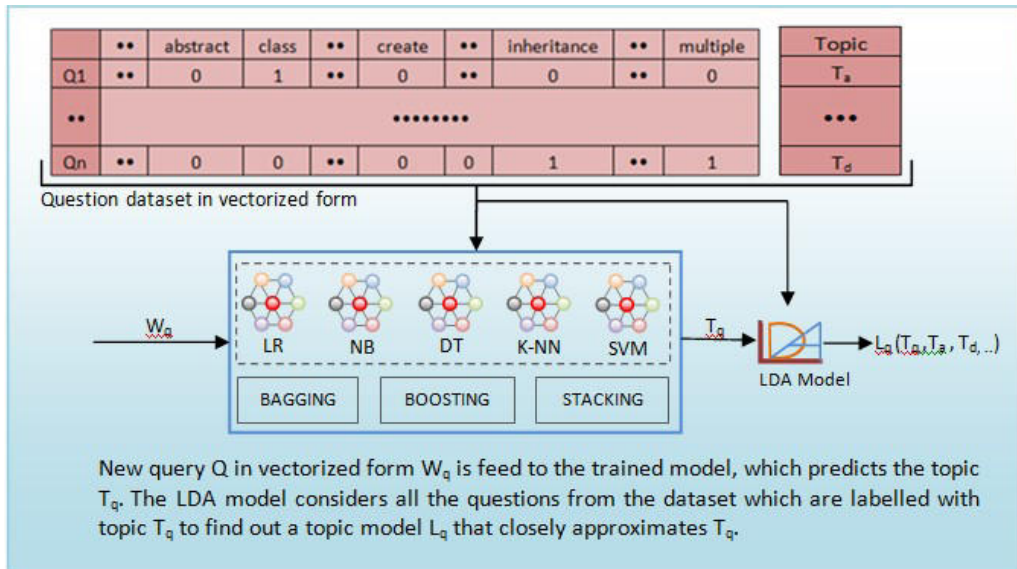


Fig. 3.2: Query to topic model

into predefined topics using a handcrafted dataset, and second, it takes a user’s query to predict a topic using the trained model and subsequently creates a dynamic topic model for the query using LDA. The preparation of the dataset involves subject experts correctly labeling the questions. This work uses a dataset of questionnaires from an elementary course on “Object-Oriented Programming with Java”. A total of 1377 questions are selected, and each question is hand-labeled with a topic name from a predefined set of 13 topics. Three subject experts have volunteered for this work. The text data is preprocessed using conventional NLP methods, viz. tokenizing, noise and stop word removal, and stemming. Then the preprocessed text is feature-extracted and vectorized using the Term Frequency - Inverse Document Frequency (TF-IDF) embedding with a one-gram. A set of machine learning models like logistic regression (LR), NB, DT, KNN, and SVM with different sets of hyper-parameters in K-fold cross-validation is used for the classification task. It explores multiple machine learning classifiers and ensemble methods. The classification models are trained over a data set that is performed to find out the best combinations of hyper-parameters. Besides individual models, ensemble methods like Bootstrap Aggregating or bagging, boosting, and stacking are also adopted for classification. For an unseen query, the machine learning classifier first predicts a topic from the predefined topic list. Then we collect all the questions labeled to that topic in our dataset. This pool of questions is fused to generate a topic model dynamically. Since our postulation is that questions in the dataset may also have latent topics besides the one assigned to them, we employ LDA to unveil those topics. We consider this topic model as the representation of the given query in the query block and is used to compare with the topic model of the LO block. Figure 3.2 shows the workflow of transforming the query to a topic model.

In the LO block, first, a repository is prepared by web-scraping three different e-learning portals [17][15][12]. From each portal, a set of interconnected LO is fetched and stored in the LO dataset. Each LO is textual content, which is represented by a single topic name as appeared in the learning portal. However, this topic name is replaced in the dataset by a dynamically created topic extracted by LDA. It enables us to counter the problem that arises due to different topic names for the same content across different portals. The textual contents are preprocessed and vectorized in a similar way as done for query text. The working mechanism of transforming LO to a topic model is shown in Figure 3.3.

Finally, for assessing the similarity between the two topic models produced from the two blocks, a similarity measuring algorithm is defined. LO that has the topic model with the highest average similarity with the query topic model is selected as the most appropriate LO for that query. In the next section, the core components of

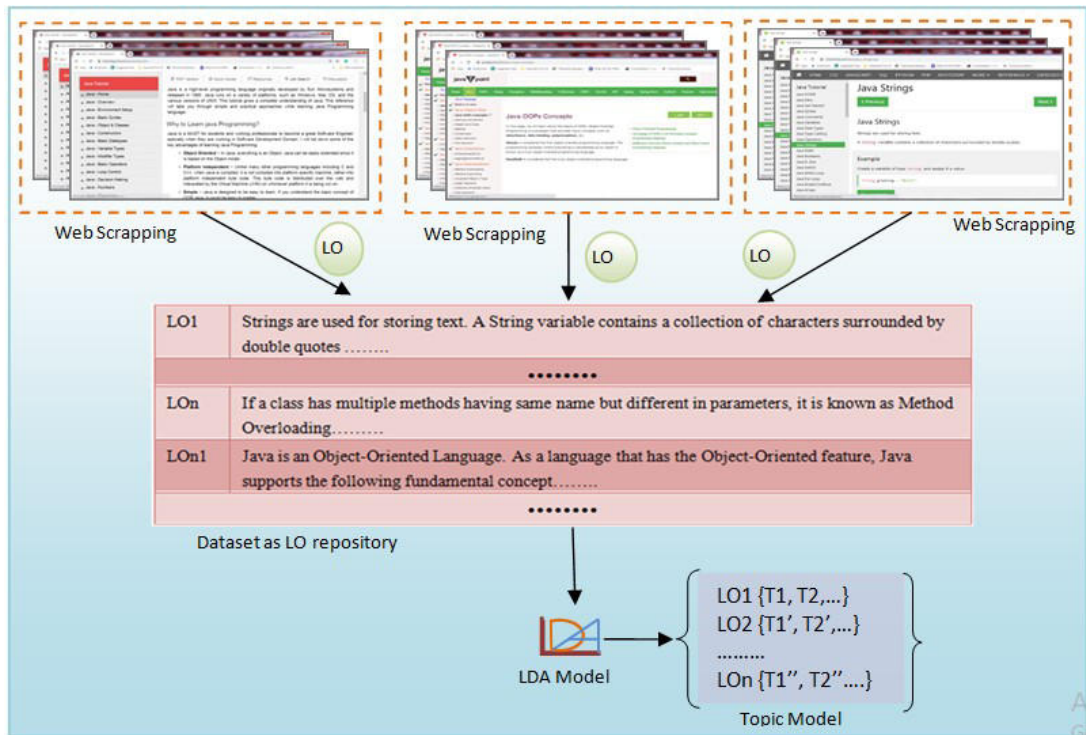


Fig. 3.3: LO to topic model

the methodology are discussed in detail.

3.1. Data Preprocessing. Common data preprocessing steps are carried out both on questions and scrapped web contents, which involve a) data cleaning and noise removal, b) stop word removal, and c) stemming. Since the text data for LO is web scrapped, it contains noise in the form of missing HTML tags, misspells, and non-ASCII characters and thus needs special attention for cleansing and removing noise. Noise removal is carried out using regular expressions features of the Python language. Next, punctuation and stop words also were removed as they do not add any information to the text analysis. Stemming removes morphological affixes from words, leaving only the root words. PorterStemmer, an NLTK library, was used for stemming in this work.

3.2. Feature Extraction. Considering the fact that users often rely on individual words rather than patterns or sequences of words in a query, we used one-gram features instead of n-gram. This also enables us to counter the problem of the unstructured query. However, it also brings a compulsion to create the one-gram model for the LOs as well. Next, for vectorization and word embedding, the TF-IDF model is adopted. Since the work has no special requirement to retain the semantic relationship between the words, TF-IDF is an inevitable choice where we can also set the minimum and maximum frequency of a word to be considered.

3.3. Machine Learning Models. Five models - LR, NB, DT, KNN, and SVM are explored for the classification task. The objective of this supervised approach is to train a model to classify any questions into some predefined topics. This work does not consider multi-label classification, i.e., each question in the database is mapped to one topic only. For fine-tuning the hyper-parameters of the models, K-fold cross-validation is performed. In addition to these five models, three types of ensemble methods, namely bagging, boosting, and stacking, are employed on top of the baseline classifiers. This work explores random forest (RF) and an ensemble of all other single models for bagging, Adaptive Boosting (AdaBoost) and extreme Gradient Boosting (XGBoost) for boosting, and KNN and SVM classifiers for stacking. The classification performances of all the

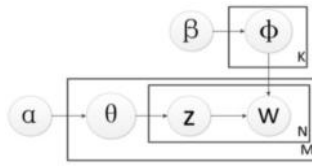


Fig. 3.4: LDA model for topic-word distribution(adopted from [2])

models are measured in terms of accuracy (ac), precision (pr), and recall (rc).

3.4. LDA Topic Model. LDA is a generative probabilistic model that can create a topic model from a given text corpus that explicitly represents a document. Each topic is modeled as an infinite mixture over a set of topic probabilities. The underlying principle of LDA is given in Figure 3.4. Where, α is the per-document topic distribution, β is the per-topic word distribution, θ is the topic distribution for document d , $P(z = k | d)$ is the probability of topic $z = k$ given document d , ϕ is the word distribution for topic k , $P(w = v | z = k)$ is the probability of word $w = v$ given topic $z = k$, Z is the topic for the n^{th} word in the document, W represents each word in the corpus.

We employed two different configurations of LDA in two phases. Since the vocabulary size of the learning content is much larger than the vocabulary size of the question set, the value of k is proportionally adjusted. For query modelling, we set the $k = 5$ for each topic, and in the LO block, the size of $k = 8$ is used.

3.5. Similarity Measure. Once the topic model for the query and the topic models for the LOs are prepared from the two blocks, the final step of the methodology is to measure the similarity between the topic models in order to identify the most appropriate LO for the learner's query. If t_q is a topic model for the user query q , we need to find out the most similar topic model t_l from the available LO topic models $(t_{l1}, t_{l2}, \dots, t_{ln})$. The simplest way of measuring this could be finding the cosine difference between the BOW representation of t_q and t_{li} . However, the conventional way of representing a topic vector in the entire word space is not very useful, as the number of words in the LO vocabulary is much larger than the number of words in one topic. This would result in each topic being a highly sparse vector. Therefore, assuming the LDA-generated topic models $(t_{l1}, t_{l2}, \dots, t_{ln})$ as a good summarization of the entire content of the LO repository, we use only the set of words used in the topics $(t_{l1}, t_{l2}, \dots, t_{ln})$ as the new word vocabulary $W = \{w \text{ for each } w \text{ in } t_{li}\}$. Thus, in this new vector space, t_q and t_{li} are represented uniformly, and the cosine distance could be used to measure the similarity between them. A similarity score of a LO is considered as an average similarity of all topics of the given LO. The LO with the highest average similarity score is taken as the best match for the query. Algorithm 1 depicts the process of finding the similarity between the topics t_{li} and t_q .

4. Results and Discussion. Five machine learning classifiers are tested on K-fold cross-validation data with accuracy, precision, recall, and F1 score as the measures of performance. Table 4.1 shows the comparative performance analysis of different single and ensemble models. LR classifier with penalty = 'l2' and solver = 'lbfgs' and SVM classifier with penalty = 'l2', kernel = 'rbf', and loss = 'squared_hinge' showed better performance than NB and KNN. We used Gaussian NB and KNN with $K = 5$, but they showed poor validation accuracy. DT with splitting_criteria = 'gini' showed a moderate performance on validation data. However, DT and NB showed good improvement when employing ensemble methods on top of the base classifiers. Bagging method RF showed an improvement of 4.06% over DT. NB with bagging increased accuracy by 1.5% compared to the baseline. However, ensemble methods do not reflect any noticeable improvement to the base models in all other cases. SVM and LR both showed good accuracy, consistent with any ensemble method. The comparative study reveals that the SVM performed better than other models at different k folds and different random states.

On the unsupervised side, topic models are built both on the questions dataset and the LO repository. Table 4.2 shows some of such LDA topics. The number of topics in the LDA model for questions and LOs is set to 10 and 15, respectively, while the word length is set to 5 and 8, respectively. Table 4.3 depicts the cosine similarity score of topic_9 (of Table 4.2) with some of the LO topics as per the proposed algorithm. As cosine similarity is a bitwise operation, the time complexity of the algorithm is $O(k * m * n)$, where, k is the number

Algorithm 1: Finding the most similar topic t_{li} for the topic model t_q
Input: t_q =topic model for user's query input LOS = set of learning objects t_l = topic model for LO
Output: best_matching_LO
for each LO in LOS { LO_avg_score=0 LO_score=0 for each t_{li} in t_l { cosine_sim_score=0 for each topic t_j in t_q { cosine_sim_score = cosine_sim_score + cosine_sim(t_{li}, t_j) } avg_cosine_sim_score = cosine_sim_score/no_of_topic_in t_q LO_score = LO_score + avg_cosine_sim_score } LO_avg_score = LO_score/no_of_LO } best_matching_LO = LO with maximum LO_avg_score

Table 4.1: Performance of Machine Learning Models.

Models	ACC	PR	RC	F1
Logistic regression (LR)	87.86	91.86	86.86	89.29
Decision tree (DT)	76.95	79.95	74.95	77.37
Naïve Bayes (NB)	82.99	85.99	81.99	83.94
SVM	88.94	92.94	87.94	90.37
KNN	68.18	71.18	67.18	69.12
Adaboost	67.58	70.58	66.58	68.52
XGboost	78.62	81.62	76.62	79.04
Stacking LR, DT, NB with meta_classifier SVM	62.03	65.03	60.03	62.43
KNN + Bagging	68.94	72.94	67.94	70.35
LR + Bagging	87.98	90.98	86.98	88.94
RF	80.08	84.08	78.08	80.97
NB + Bagging	84.23	87.23	83.23	85.18
SVM + Bagging	89.04	93.62	87.94	90.69

of LO, m is the number of topics in the question topic model, and n is the number of topics in the LO topic model.

5. Conclusions. This paper presents a methodology for an automated recommender system that maps a user's query to an appropriate LO that covers the required knowledge for that query. It relies on dynamic topic modelling using LDA to relate between the query and the required LO. Different machine learning models like LR, NB, DT, KNN, and SVM are trained on a hand-curated dataset of questionnaires labeled with some predefined topics. Ensemble methods like bagging, boosting, and stacking are also tested along with the baseline classifiers. SVM outperformed the other models in terms of prediction accuracy, precision, and recall. Given a learner's query, the prediction model can predict a topic from a set of predefined topics. Subsequently, a dynamic topic model is created using LDA from the questions in the dataset on that topic. On the other side, topic models are created from the LOs fetched from e-learning portals. Then, an algorithm is proposed to find out the proximity between two topic models. The LO corresponding to the topic model that has the highest average similarity score with the topic model of the user query is considered the most appropriate LO for that query. This work does not employ any sequential or temporal model for considering textual data,

Table 4.2: LDA Topic Models.

LDA model on	Topic_4		Topic_6		Topic_9	
	Word	Prob	Word	Prob	Word	Prob
Questions dataset	Memory	0.059	exception	0.110	class	0.050
	Thread	0.045	block	0.096	method	0.038
	Heap	0.044	catch	0.060	interface	0.029
	Stack	0.035	try	0.043	java	0.029
	Java	0.030	finally	0.035	abstract	0.028
LO repository	Topic_2		Topic_7		Topic_11	
	Word	Prob	Word	Prob	Word	Prob
	variable	0.072	method	0.026	class	0.142
	method	0.029	java	0.024	method	0.033
	instance	0.029	string	0.023	abstract	0.020
	class	0.028	class	0.020	java	0.017
object	0.027	value	0.019	inner	0.017	

Table 4.3: Similarity Scores

LDA topic from Java question corpus	LDA topic from LOs from web portals	Cosine similarity score
Topic_9: 0.050*“class” + 0.038*“method” + 0.029*“interface” + 0.029*“java” + 0.028*“abstract”	Topic_2: 0.072*“variable” + 0.029*“method” + 0.029*“instance” + 0.028*“class” + 0.027*“object”	0.4
	Topic_7: 0.026*“method” + 0.024*“java” + 0.023*“string” + 0.020*“class” + 0.019*“value”	0.6
	Topic_10: 0.142*“class” + 0.033*“method” + 0.020*“abstract” + 0.017*“java” + 0.017*“inner”	0.8

assuming users’ queries consist of fewer words and are often unstructured. Therefore, the future scope of this work is to explore word embedding techniques like word2vec and temporal deep learning models like LSTM and Transformer for the prediction task.

REFERENCES

- [1] I. BHATTACHARYA AND J. SIL, *Query classification using lda topic model and sparse representation based classifier*, in Proceedings of the 3rd IKDD Conference on Data Science, 2016, 2016, pp. 1–2.
- [2] D. M. BLEI, A. Y. NG, AND M. I. JORDAN, *Latent dirichlet allocation*, Journal of Machine Learning Research, 3 (2003), pp. 993–1022.
- [3] S. DAS AND R. CHATTERJEE, *Work in progress - an approach for mapping of auto generated questions to related topics in the curriculum*, in International Conference on Advances in Computing, Communications and Informatic, 2015.
- [4] B. L. ERICKSON, C. B. PETERS, AND D. W. STROMMER, *Teaching First-Year College Students - Revised and Expanded Edition of Teaching College Freshmen*, Jossey-Bass, San Francisco, CA, revised ed., 2006.
- [5] E. ESLINGER, B. WHITE, J. FREDERIKSEN, AND J. BROBST, *Supporting inquiry processes with an interactive learning environment: Inquiry island*, Journal of Science Education and Technology, 17 (2008), pp. 610–617.
- [6] Z. GULZAR AND A. A. LEEMA, *Course recommendation based on query classification approach*, International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 13 (2018), pp. 69–83.
- [7] X. LI AND D. ROTH, *Learning question classifiers*, in COLING 2002: The 19th International Conference on Computational Linguistics, 2002.

- [8] E. LOPEZ, *Mckeachie's teaching tips: Strategies, research, and theory for college and university teachers*, The Review of Higher Education, 27 (2004), pp. 283–284.
- [9] S. PAL, P. K. D. PRAMANIK, AND P. CHOUDHURY, *Enhanced metadata modelling and extraction methods to acquire contextual pedagogical information from e-learning contents for personalised learning systems*, Multimedia Tools and Applications, 80 (2021), pp. 25309–25366.
- [10] S. PAL, P. K. D. PRAMANIK, A. MAITY, AND P. CHOUDHURY, *Learner question's correctness assessment and a guided correction method: enhancing the user experience in an interactive online learning system*, PeerJ Computer Science, 7 (2021), p. e532.
- [11] S. PAL, P. K. D. PRAMANIK, A. NAYYAR, AND P. CHOUDHURY, *A personalised recommendation framework for ubiquitous learning system*, in Proceedings of the 6th International Conference on Intelligent Information Technology (ICIIT '21), 2021, pp. 63–72.
- [12] J. T. POINT, *Java tutorial*, 2011. June, 2023.
- [13] B. QU, G. CONG, C. LI, A. SUN, AND H. CHEN, *An evaluation of classification models for question topic categorization*, Journal of the American Society for Information Science and Technology, 63 (2012), pp. 889–903.
- [14] A. SINGH AND K. VISWESWARIAH, *Cqc: classifying questions in cqa websites*, in Proceedings of the 20th ACM international conference on Information and knowledge management, 2011, pp. 2033–2036.
- [15] TUTORIALSPPOINT, *Java tutorial*, 2013. June, 2023.
- [16] N. VAN-TU AND L. ANH-CUONG, *Improving question classification by feature extraction and selection*, Indian Journal of Science and Technology, 9 (2016), pp. 1–8.
- [17] W3SCHOOLS, *Java tutorial*, 1999. June, 2023.
- [18] D. A. WILEY, *The instructional use of learning objects*, vol. 1, Agency for Instructional Technology and Association for Educational Communications & Technology, Bloomington, Indiana, 2002.
- [19] H. YANG, Q. HU, AND L. HE, *Learning topic-oriented word embedding for query classification*, in Advances in Knowledge Discovery and Data Mining. PAKDD 2015. Lecture Notes in Computer Science, 2015, pp. 188–198.
- [20] W.-N. ZHANG, T. LIU, Y. YANG, L. CAO, Y. ZHANG, AND R. JI, *A topic clustering approach to finding similar questions from large question and answer archives*, PloS one, 9 (2014), p. e71511.

Edited by: Rajni Mohana

Special issue on: Sentiment Analysis and Affective computing in Multimedia Data on Social Network

Received: Jul 31, 2023

Accepted: Oct 5, 2023

