NLP AND IR BASED SOLUTION FOR CONFIRMING CLASSIFICATION OF RESEARCH PAPERS

[1]KHALID M.O. NAHAR, [2]NOUH ALHINDAWI, [3]OBAIDA M. AL-HAZAIMEH, [4]RA'ED M. AL-KHATIB, [5]ABDALLAH M AL-AKHRAS

[1]Department of Computer Sciences, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid-21163, Jordan

[2] Department of Software Engineering, Faculty of Sciences and Information Technology, Jadara University, Irbid, Jordan

[3] Department of Computer Science, Al-Balqa' Applied University, Al-Huson University College, P. O. Box 50, Irbid, Jordan

[4]Department of Computer Sciences, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid-21163, Jordan

[5]Department of Computer Information System, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid-21163, Jordan

E-mail: [1]khalids@yu.edu.jo, [2]hindawi@jadara.edu.jo, [3]dr_obaida@bau.edu.jo, [4]raed.m.alkhatib@yu.edu.jo, [5]abdalla_akhras@yu.edu.jo

ABSTRACT

In this paper, an approach is presented for classifying and categorizing the research's papers in very accurate manner. Typically, the papers are classified into clusters based on the concepts and the contents, this clustering process is mainly depends on the title of the paper. However, a lot of papers have ambiguous title or have a very short title. Therefore, the researcher needs to cluster and classify the papers not just depending on the title, but also include other parts of the paper like: abstract, keywords, and may be some key parts of the paper. This process is time consuming since the researchers spend a lot of time to decide the related cluster of the undertaken paper. Our presented approach provides an automatic, short time, and accurate solution, which mainly depends on Information Retrieval (IR) as core process along with some Natural Language Processing (NLP) techniques. Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing (LSI) are the two IR algorithms which used in the new approach. We use the LDA for classifying the papers using the concept of topic modeling. And we use the LSI for performing querying. The new approach uses the title of the paper, the abstract, and the keyword for performing the classification process. Two distinct experiments were conducted over 600 papers in the field of computer science. The results show the efficiency of the proposed approach in classifying and mapping the papers accurately and efficiently.

Keywords: *NLP and Information Retrieval (IR), Classification; Topic Modeling; Latent Dirichlet Allocation; Latent Semantic Indexing; Gensim*

1. INTRODUCTION

Typically, the research's papers usually have a standard template and structure that the researchers follow while writing. Most of the papers begin with a title followed by an abstract and some keywords. These three main components describe the contents of the research paper [1]. Usually, the title gives the overview of the context, the abstract summarizes the context and the keywords indicate the core concepts [2]. Therefore, the title of the papers must be chosen conceptually and abstractly and should reflect the main theme of the paper.

The readers or the researchers usually start the process of understanding any field of study or research problem by gathering some related information about them. This can be done by reading the related papers or topics [3][4]. This is usually done by looking after the related paper's titles through databases systems like Google, Yahoo, etc. The research paper title is considered to be the first part that the researcher mostly read. The title also defines the research study field, topic, or domain.

Many researchers gave a set of recommendations for choosing paper titles. For instance, avoid including unnecessary words in the paper title such as, "*A Study to Investigate the...*," or "*A Review of the....*”. These phrases are obvious and generally unnecessary. On the other side, choosing a short title is also not preferable due to the lack of meaningful knowledge we gain. Moreover, the short title, for example, a paper with the title, "*World Temperature*" is so non-specific, it could be the title of a book and may possibly observe a wide of information related to the temperature of the world. Moreover, it is considered to be ambiguous title. A good title should provide information about the focus of your research study.

In general, all database systems have an interface for the users and the researcher which enable them to write the query that describe what they looking for using the natural language. Typically, the result of the query is a list that contains the relevant papers or documents ranked based on their relevancy to the user query. This relevancy is computed based on the similarity between the user query and the databases documents. The retrieved list would be manually investigated by the reader and would be decreased based on reader’s judge for the relevancy between the papers titles in the retrieved list and between what they looking for. Thus, the authors must choose the title of their papers carefully. One of the most popular problems in many of the research papers is the unsuitability of the papers title with the subject and the contents of the papers. A lot of authors chose a title for their paper based on either part of the algorithms of the methodology they use, or based on their knowledge, or based on their major of study. This paper presents a solution to the problem of classifying the papers titles conceptually by utilizing information retrieval and topic modeling in the classification process. Topic modeling have been utilized recently by a lot of researchers to solve, examine, and analyze many problems in the field of software engineering, especially when dealing with textual information like source code and developers commits, etc.. Recently, the usage of topic modeling were notably increased, therefore, many different toolkits have appeared for the wide functions of topic modeling such as Gensim [5]. In this paper, we use the Gensim as it is a free Python library that is aimed to automatically extract the semantic topics from a set of documents. This paper investigates the usefulness of LDA and LSI in classifying research’s papers.

Moreover, another important application of the topic modeling which appear recently, it includes social media networks such as Facebook, Myspace and Twitter. These websites become a very important tools for social communication and a very important source of social information. Breaking news, eyewitness accounts and organizing large community of users are done using these websites across the globe.

These websites motivate researchers to use messages posted by users to infer users’ interests, model social relationships, track news stories and identify emerging topics. Researchers apply the principles and techniques provided by topic modeling which is based on LDA, SVD, and LSI to model the user interest and needs.

Going farther than modeling interest and needs, is the firm classification of the research papers that is to be done and enhance in this research.

Usually, a question always raised when writing a new research paper is that "*Are the referenced papers used in this research are strongly related to the topic or problem under study?*". Therefore, we are motivated to do this research in order to correctly identify and classify the referenced research papers so that they reflect the main theme of the research paper. This paper try to answer the following research question (RQ):

RQ1: is the proposed approached efficient and accurate in classifying any research area’s papers?.

The rest of paper is organized as follows. In section 2, we present the related work. Section 3 describes the proposed methodology. In section 4, we describe Latent LDA and LSI algorithms. The conducted experiments are presented in section 5. The results are discussed in section 6. Finally, Section seven summarizes the conclusions and briefly highlights future work followed by the references.

2. RELATED WORKS

This section presents an overview about topic modeling usages along with main application for LDA and LSI models. Topic modeling is earning increasing attention in several text mining communities [6] [7].

Zhang et al. (2007) introduced a model to mix LDA into a community-detection process [8].

Chang et al. (2009) presented a probabilistic topic model to analyze text corpus and deduce descriptions of the entity and of relationships among that entity on Wikipedia [9]. While Wang et al. (2012) suggest a new method of constructing a strong feature thesaurus depending on LDA and information gain models [10] [11]. Meanwhile, at the same year i.e 2012, Kim et al. (2012) presented a model called a language independence semantic which can compare and links between the related document without using any tag or databases [12].

On other side, a new algorithm called LSI was shown to perform better than the simple word N-gram feature vectors in [13][14], where different kinds of vector similarity metrics are used (e.g., count vectors, Jaccard, cosine, and distance measure) have been evaluated. Due to the high computational cost of LSI, there have been many works around the area of approximate matrix factorization; these algorithms maintain the spirit of Singular Value Decomposition but are very easier to calculate [15].

Moreover, a sentiment classification using Machine Learning Techniques was also presented by [16] [11]. The authors examined the effectiveness of applying three machine learning algorithms, namely, Naive Bayes, and the maximum entropy. The results were promised and good when comparing to the results of classification manually by human.

In [17], the authors used LDA along with Hellinger Distance in order to confirm the quality of software documentations. The approach depends mainly on building two distinct corpuses, one for the source code fragments, where the other one is for the external documentation. The similarities then were computed between the topics.

A visualization process was also added to the process of classification in [18], the authors tried to reduce the complexity of domain problem using the visualization concept.

The authors of [19], introduced a new supervised approach for classifying scientific research papers by using analysis of the paper's interrelationships. The authors exploited links such as citations, common authors, and common references to allocate subject to papers.

Based on what we have discussed in the previous works, no one think of using the principles of topic modeling to enhance new research papers generation. These could be achieved by accurately classifying and categorizing the referenced research papers related to the topic or problem under Study.

Consequently, we will improve the LDA model and LSI model to present a good model for classifying published research papers in the field of information technology. Then, the user can search, browse and summarize large archives of information technology papers.

3. PROPOSED APPROACH

As we declared before, the proposed approach is mainly depends on LDA and LSI algorithms. The proposed approach is done in a pipeline structure. In other words, a set of steps are done sequentially and the output of one step is considered as input for the next one.

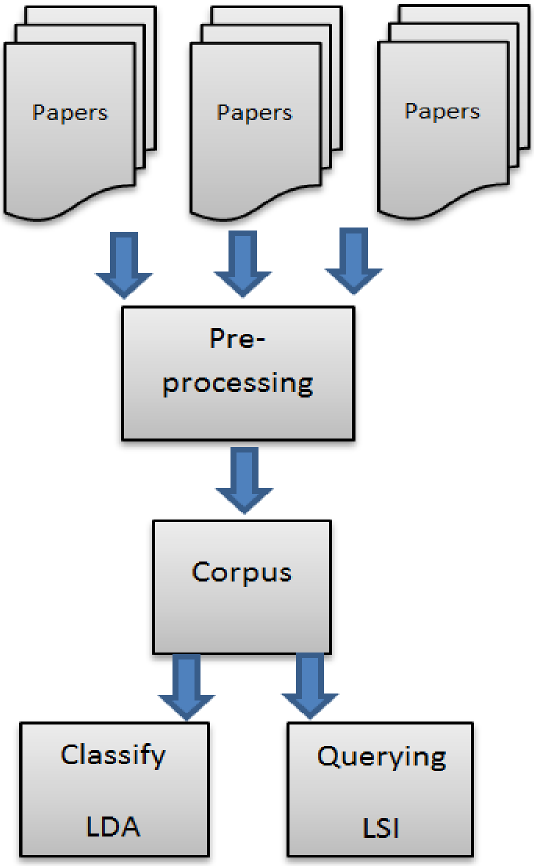

Figure 1: Steps of the proposed approach

The following steps describe the proposed approach as shown in Figure 1:

Step 1: the initial step is collecting the set of papers that need to be processed, in this step; we choose the title of the paper, keywords, and the abstract. All selected are in the field of computer science and software engineering.

Step 2: a set of preprocessing steps are done here, the stop word, special marks, and punctuation marks are removed here. Moreover, the stemming process is done here using porter stemmer in order to extract the root of all words.

Step 3: as a next step, we build a corpus for the collected papers, in the corpus; each paper will have a correspondence document.

Typically, the above preprocessing steps are crucial and required. It reduces the indexing size, increases the results precision. We use LDA here to extract the main topics of the undertaken system, the number of topics is desired by the user, and it depends on the size of the corpus and on the domain and concept of the corpus. Each topic Ti contains a set of related terms; those terms are ranked on the topics based on the relevancy to the topic concepts. In other words, the Ti represents a distinct likelihood allocation that defines how expected each word is to come out on a certain topic. In the following subsections we will talk in more details about LDA and LSI consequently.

3.1 Latent Dirichlet Allocation (LDA)

LDA is a probabilistic topic models that is used to find the number of occurrences for each words in targeted documents [1]. Further, the LDA utilizes the powerful of latent variables to extract the occurrence patterns of specific words in documents (corpora) then compare them with other data. LDA is considered a powerful generative probabilistic model for counting the occurrences of words and collections of discrete data from text corpora. The main process of LDA is to generate a random mixtures representation for documents over latent topics, then classifying and clustering each topic according these words distribution.

The basic representation of LDA model is as described in [20], where many (K) topics $\emptyset_{\boldsymbol{k}}$, when there is, $k \in \{1, \ldots, K\}$ implies that the discrete distributions in topics are distributed over words. For instance, suppose we have a topic on "*Sport*", with relevant probabilities assigned to Bag of Words (BoWs) like "*gym*", "*player*", "*football*", and "*stadium*". The workflow process of LDA topic models can be illustrated in the following pseudo code:

```
Generate each topic $\emptyset_{\boldsymbol{k}} \sim Dirichlet(\eta), k \in \{1,\ldots,K\}$
  For each document $j$
    Generate a distribution over topics $\boldsymbol{\theta}_j \sim Dirichlet(\boldsymbol{\alpha})$
      For each word $i$ in document $j$
        Sample a topic $z_{ij} \sim Dirichlet(\theta_j)$
        Sample a topic $w_{ij} \sim Dirichlet(\emptyset\, z_{ij})$
```

Therefore, the LDA will be inference to very big benchmark and datasets. The process of LDA workflow can be explained with following exemplar of documents and topics model.

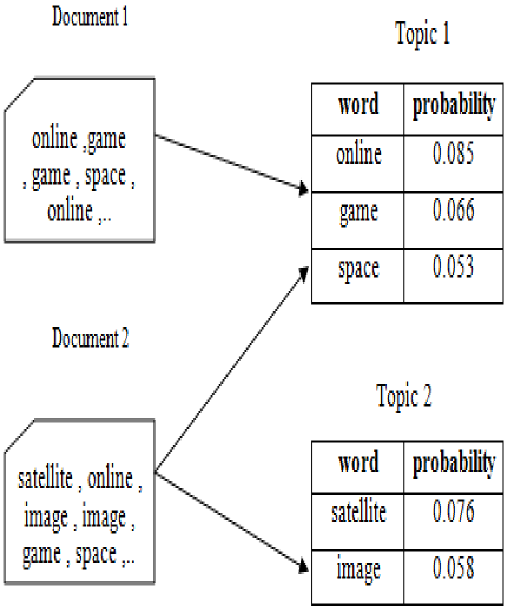

Figure 2: Illustration of LDA model

For example, in Figure 2 Document 2, is half about Topic 1 (50%) and half about Topic 2 (50%), while Document 1 is only about Topic 1 (100%). Each topic is represented as a probability distribution over a controlled vocabulary, usually all the words appearing in the document collection. In our example, Topic 1 has words like "*online*" (8.5%), "*game*" (6.6%), and "*space*" (5.3%) with high probability and Topic 2 has words like "*satellite*" (7,6%), "*image*" (5,8%) with high probability. Given this information, we could label Topic 1 as "*online game*" and Topic 2 as "*satellite imaging*". Consequently, we could say that Document 1 is purely about "*online game*", while and Document 2 is a mix of the "online game" and the "*satellite imaging*" topics.

These procedures shown in Figure 3 are the preparation of the corpus for LDA. Firstly, all the words in the documents are assigned a unique id by using a dictionary. Then, the documents are converted to vectors. And finally, the similarities between the vectors are measured. The speed of building the corpus is depending on the size of the original documents. In other words, the LDA takes a chunk of documents, process it and keeps running iteration until it reaches a certain amount of iteration [21]. So, the corpus building process is depending on the size of the documents. However, the speed of classification is relates to the implementation of the classifier itself.

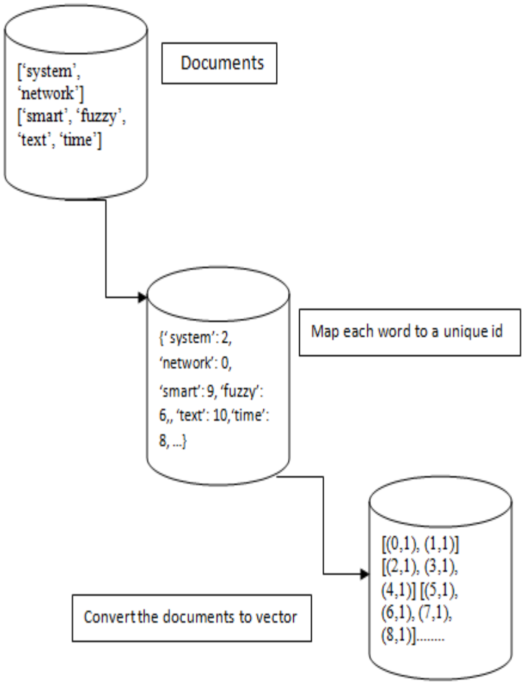

Figure 3: Stages of Corpus Preparation for LDA

3.2 Latent Semantic Indexing (LSI)

The LSI is a powerful statistical technique information retrieval in textual representation domain. Mainly, the LSI is used lately to solve lexical matching problems, and is considered one of the best tools used to maintain the semantic information between the words with high accuracy [6][22]. Therefore, LSI is useful in many applications like: author recognition, plagiarism detection, search engines, and text similarity. Typically, LSI projects, runs queries on documents in a space with semantic dimensions well-known as "latent". Latent is statistically adapts the conceptual indices instead of individual words in the process of retrieval [20][23]. Consequently, the main contexts of LSI is to examine if a specific word exists or not by applying the similarity process through the documents. Finally, the behavior of LSI model is similar to human learning when people learn the language of mother tongue to acquire a new vocabulary.

The retrieval process of LSA is basically used the truncated singular value decomposition (SVD) algorithm [24][23]. SVD works to estimate the structure in word usage in over all the documents, if there are some underlying or latent structure at these words usage, they are partially obscured or not clearly estimated. Therefore, the retrieval process is done by utilizing well-prepared database for the previous singular values and vectors that are obtained by truncated SVD algorithm. The executable run illustrates more robust indicators of meaning with better performance to these processes of derived vectors against the individual one.

The main steps to build the process of LSA model are: i) Preprocess the collected documents by stemming, splitting composite words, and removing the stop words, ii) building the frequency matrix, which is performed by constructing the term-document matrix (TDM), iii) applying the weight functions in order to increase the efficiency of the information retrieval process, and to help to allocates weights to the terms based on their occurrences. This weight functions work to replace each element with the product of a Local Weight Function (LWF), and the Global Weight Function (GWF). While the LWF responsible to find the frequency of a specific word within a particular text, where GWF inspects a term's frequency in over all the documents, iv) decomposing the initial variables, and v) start project queries of LSA for information retrieval.

LSI grows from the problem of how to discover relevant documents from search words. The fundamental difficulty grows when we compare words to find relevant documents because what we want to do is comparing the meanings or concepts beyond the words. LSI attempts to solve this problem by mapping queries into a large document and doing the comparison in this space.

3.3 Singular Value Decomposition (SVD)

SVD is considered an ancient numerical analysis technique that was discovered long ago to serve many applications [25]. Beltrami and Jordan in the 1870's introduced SVD and used it for real square matrices. After that, and in 1902 Autonne's used it for complex matrices [26] [27]. However, SVD was improved by Eckart and Young Later, in 1939, to include rectangular matrices as cited [26]. Recently the SVD becomes one of the most important numerical techniques used in image processing applications, such as image watermarking, image hiding, image compression and noise reduction [25] [26]. SVD could be applied on any medium. But mostly, it is famous in image processing. The widespread use of SVD is due to its important features and characteristics, which are the following [25] [26][28]:

1. Good stability of the singular values S A of any image, i.e., no significant changes to the SVs of images will occur upon the addition of small perturbations.

2. The singular values (S) of an image specify its algebraic properties, representing an image's luminance, whereas the singular vectors (U and V) represent the geometry properties of an image.

3. Singular values are in descending order, and many of them have small values compared to the first singular value. Updating or ignoring these small singular values at the reconstruction stage leads to a slight and negligible effect on an image's quality.

4. SVD can be applied on square or rectangle matrices.

4. EXPERIMENTS SETUP AND RESULTS

In this section, we describe the data set and the taken steps while running the experiments along with the results and we finish by discussing the results and the evaluation.

4.1 Data Acquisition

As we explained early in this paper, for the experiments, we collected 600 research's papers to be used. The papers are related to the field of computer science and software engineering. And consequently, we built a corpus; the corpus includes a set of corresponding documents for the 600 paper. Each document contains a description for a paper; we chose the title, keywords, and the abstract for each paper. Once these steps are completed, the corpus became ready to be used as input for Gensim.

4.2 Experiments & Discussion

Gensim is a free tool which was implemented in Python, it contains an implementation for Singular Value Decomposition (SVD), and it has an implementation for LDA and LSI modules. We used Gensim to predict the topic distribution for each document. The topics distributions are utilized for both training and testing the classifier and evaluating the results.

As mentioned before, two separate experiments are conducted in this paper; the first one is done by using LDA to extract the topics. For our experiment we extracted 20 topics and we chose 10 terms for each topic to appear. The second one is done by using LSI in order to run a query. The query here is a title or keywords of new paper, and the results of running the query is a list. This list contains the relevant documents from our corpus that close to our query ranked increasingly.

Table 1: Samples of the extracted topics with top five related papers terms.

Topic #	Term 1	Term 2	Term 3	Term 4	Term 5
1	Cloud	Compute	link	Network	Security
2	Image	Process	Decrypt	Key	Matrix
3	Feature	Query	Select	Rank	Retrieve
4	Examine	Test	Enhance	Static	Faults
5	Security	Complex	Distribute	Intrusion	Cloud

For the first experiment, we built the corpus as mentioned before, and we use the dimensionality of the singular vale decomposition (SVD) to be 200. We extracted 10 topics from the corpus, where each topic has top 15 related terms. We used 50% as threshold for selecting each topic's terms. As shown in Table 1, each topic of the extracted topics includes the terms of the papers that have a similar subject in the field of computer science and software engineering.

We performed a manual inspection over the extracted topics. The inspection process is done by two master students of computer science. Each student separately analyzed the papers of each topic and gave a score for the relevancy of each paper in the selected topics.

Table 2: The results of inspection for student number one.

Topic #	*Score 3*	*Score 2*	*Score 1*	*Score 0*
1	6	3	1	0
2	7	2	1	0
3	5	4	1	0
4	8	1	1	0
5	9	0	0	1
6	6	2	2	0
7	5	5	0	0
8	4	6	0	0
9	7	3	0	0
10	8	1	1	0

The inspection process includes many steps that taken from both of the student, they read the paper abstract and the methodology of each paper to come up with accurate results and validation. The score is given as follow, score 3 means strong relevancy, score 2 means normal relevancy, score 1 mean weak relevancy, and 0 means out of scope paper (irrelevant). Table 2 and Table 3 represent the results of inspection for both students.

Table 3: The results of inspection for student number two.

Topic #	*Score 3*	*Score 2*	*Score 1*	*Score 0*
1	8	2	0	0
2	4	5	1	0
3	6	1	3	0
4	7	3	0	0
5	5	2	2	1
6	5	3	1	1
7	7	3	0	0
8	5	5	0	0
9	6	2	1	0
10	9	0	1	0

The tables below represent the score examined for each topic, for example, in Table 2, for topic number 1, there are 6 papers have score 3, and 3 paper as score 2 and one paper with score 1. Also, there is no paper out of scope of the topic. This means that all papers in topic number one are related and this confirms the efficiency of the proposed approach. As we see in Table 2 and Table 3, both of the two students filtered the extracted topics individually, and as shown the results are close in both tables. The results for student number one are also shown in Figure 4.

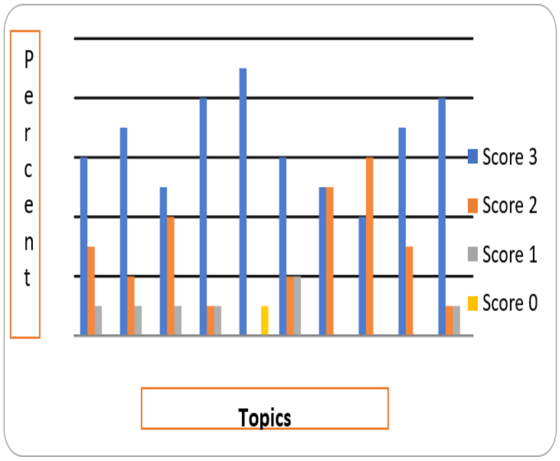

Figure 4: Student number one evaluation

By analyzing the set of extracted topics, we can examine if we chose our data for some period of time the new trends in research and how the researcher's directions are in solving a problem in a specific field or subject. Moreover, it may become possible to find out what the researchers are care and concern about. The results show that the undertaken papers have been linked to the right category. As we see in Table 1, there are five different topics. For instance, topic number one is about cloud computing, 124 out of 600 papers are categorized with cloud computing subject. And as seen the first two terms in the topic are cloud as the first term and the computing term as the second one. These results constitute the efficiency of the proposed approach. Table 4, represents the subjects of the papers the proposed approach.

Table 4: Results of classifying the 600 papers.

Subject	**Number of papers**
Cloud Computing	124
Image Processing	67
Software Development	53
Software Comprehension	42
Network Security	46
Machine Learning	115
Natural Language Processing	65
Mixed subjects	109

We conducted an experiment as an advanced level of classification for papers which have been classified as software development. The new experiment is to further classify the papers into the sub-subjects. Table 5 represents the new classification for the 53 software development papers.

As shown in Table 5, the second level of classification give more accurate labeling for the paper's subjects. This clearly shown in the Table 5, we found 8 sub-subject in the field of software that the undertaken papers belong and relate to, and we found that there are 5 papers which have mixed topic. The mixed topics papers are the papers that have employed many different module, contribution, tools, etc.

4.3 Evaluation & Analysis

We conducted an experiment for papers subject matching for new paper to the subject it relates to. Here, we use LSI to accomplish this target. A set of steps are taken here, the first one is to build the corpus as we did in the previous subsection. As a next step, we build the LSI space. Afterward, for any given new paper, we take the title of the paper as a query. Finally, we run the query over the corpus using LSI, and an ordered list of the relevant papers to our query is retrieved. And here, the role of researcher is raise to investigate the retrieved list matching with his query.

We used two evaluation standard metrics in IR of the text classification, which are Recall and Precision. The meaning of TP, FP, Precision, Recall measurements, are detailed in Table 5.

Table 5: The summary of measurements (Precision and Recall) used for evaluation.

The Measurement	The Meaning
TP	Number of true positives (instances correctly classified as a given class).
FP	Number of false positives (instances falsely classified as a given class).
FN	Number of incorrect classification of positives instances.
Precision (p)	Proportion of instances that are truly of a class divided by the total instances classified as that class
Recall (r)	Proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP rate).

The Precision (p) measure is computed based on Eq. (1).

$$p = \frac{TP}{TP + FP} \tag{1}$$

The Recall (r) measure is computed based on Eq. (2).

$$r = \frac{TP}{TP + FN} \tag{2}$$

Consequently, The Precision can demonstrates the number of documents is in right cluster with respect to the cluster size. However, the Recall shows how many documents are in the right cluster with respect to total documents.

Table 6: Advanced classification for the 51 software development papers.

Subject	Number of papers
Empirical software engineering	10
Software evolution & maintenance	15
Software testing	8
Mining software engineering	6
Software visualization	3
Requirements engineering	3
Program comprehension	2
Model-driven engineering	1

As shown in Table 7, the results for 12 queries are displayed. The 12 queries are built using 12 papers in the field of software evolution and maintenance. As shown in Table 6, there are 15 relevant papers in the field of software maintenance and evolution.

Table 7: Results for 12 queries.

Topic #	Recall	Precision
1	80 %	18 %
2	75 %	16 %
3	90 %	22 %
4	85 %	20 %
5	80 %	18 %
6	80 %	18 %
7	95 %	24 %
8	98 %	26 %
9	90 %	22 %
10	100 %	30 %
11	90 %	22 %
12	100 %	30 %

A pictorial view of Table 7 clearly shows the recall/precision ratios for each topic.

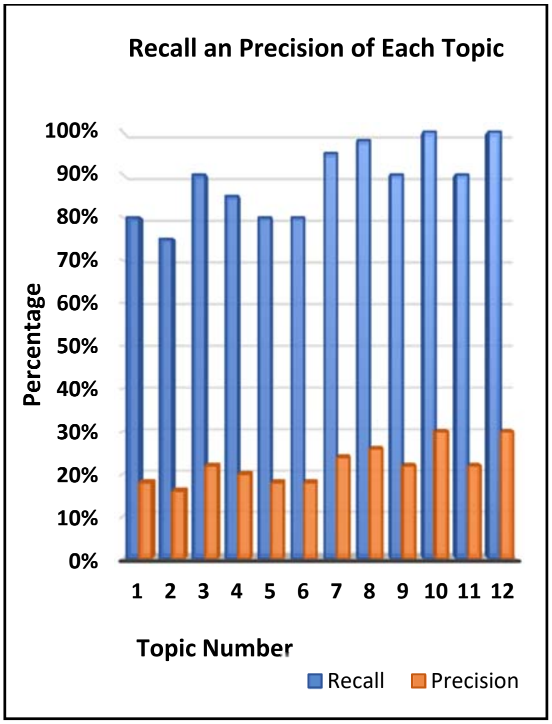

Figure 5: Student number one evaluation

In summary, comparing our approach to prior methods and approaches, we can say that our proposed approach is fully automated one. In this approach we used LDA, VSD, and LSI algorithms jointly to achieve a firm and accurate classification of research papers automatically which distinguish the approach from prior ones.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel approach which mainly depends on utilizing the LDA algorithm as a feature extractor through topic modeling and LSI as querying. The presented framework makes an important step in the direction of current trends in Natural Language Processing (NLP) and information retrieval (IR). Our framework makes a conscious effort to make parsing, processing and transforming corpus into vector spaces as intuitive as possible.

We conducted two separate experiments for research papers classification and for new research paper querying consequently. The new approach depends on Gensim, an implementation of some of the popular topics model algorithms, such as LSI and LDA in python which uses Singular Value Decomposition (SVD). The results show that the presented approach outperformed the state-of-art results and accuracy.

The RQ1 raised at the beginning of this research is achieved since for now we can efficiently and accurately classify research papers based on its topic. The whole process was carried by two efficient algorithms called LDA and LSI.

The main limitation that faces our approach is the un-availability of huge, accurate, and complete dataset that includes sufficient predetermined classes. Consequently, we build our own corpus by collecting 600 research papers in the field of computer science and software engineering.

In the future work, we plan to experiment with adding more topics to the model. We also plan to generate other categories from the texts of the descriptions that employ different basis for categorization. One candidate method for that is Named Entity Recognition, which allows for extraction of named entities, e.g. names of persons, companies, countries and titles of books.

REFRENCES:

[1] M. B. Dos Santos, “The textual organization of research paper abstracts in applied linguistics,” *Text-Interdisciplinary J. Study Discourse*, vol. 16, no. 4, pp. 481–500, 1996.

[2] B. Samraj, “An exploration of a genre set:

Research article abstracts and introductions in two disciplines," *English Specif. Purp.*, vol. 24, no. 2, pp. 141–156, 2005.

[3] N. Alhindawi, J. Alsakran, A. Rodan, and H. Faris, "A Survey of Concepts Location Enhancement for Program Comprehension and Maintenance," *J. Softw. Eng. Appl.*, vol. 7, no. 5, pp. 413–421, 2014.

[4] E. Friginal and S. S. Mustafa, "A comparison of US-based and Iraqi English research article abstracts using corpora," *J. English Acad. Purp.*, vol. 25, pp. 45–57, 2017.

[5] Gensim, "https://radimrehurek.com/gensim/tutorial.html." .

[6] K. Nahar, H. Al-Muhtaseb, W. Al-Khatib, M. Elshafei, and M. Alghamdi, "Arabic phonemes transcription using data driven approach," *Int. Arab J. Inf. Technol.*, vol. 12, no. 3, pp. 237–245, 2015.

[7] C. Jacobi, W. van Atteveldt, and K. Welbers, "Quantitative analysis of large amounts of journalistic texts using topic modelling," *Digit. Journal.*, vol. 4, no. 1, pp. 89–106, 2016.

[8] H. Zhang, C. L. Giles, H. C. Foley, and J. Yen, "Probabilistic community discovery using hierarchical latent gaussian mixture model," in *AAAI*, 2007, vol. 7, pp. 663–668.

[9] J. Chang, J. Boyd-Graber, and D. M. Blei, "Connections between the lines: augmenting social networks with text," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 169–178.

[10] B. Wang, Y. Huang, W. Yang, and X. Li, "Short text classification based on strong feature thesaurus," *J. Zhejiang Univ. C*, vol. 13, no. 9, pp. 649–659, 2012.

[11] K. M. O. Nahar and I. Alsmadi, "Information analysis to study interactions between different genes and diseases," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 1–5.

[12] K. Kim, B. Chung, Y. Choi, S. Lee, J.-Y. Jung, and J. Park, "Language independent semantic kernels for short-text classification," *Expert Syst. Appl.*, vol. 41, no. 2, pp. 735–743, 2014.

[13] M. D. Lee, Da. J. Navarro, and H. Nikkerud, "An empirical evaluation of models of text document similarity," in *Proceedings of the Cognitive Science Society*, 2005, vol. 27, no. 27.

[14] T. K. Das and K. M. O. Nahar, "A Voice Identification System using Hidden Markov Model," *Indian J. Sci. Technol.*, vol. 9, no. January, 2016.

[15] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," *Proc. Conf. Empir. Methods Nat. Lang. Process.*, pp. 79–86, 2002.

[17] N. Alhindawi, O. M. Al-hazaimeh, and R. Malkawi, "A Topic Modeling Based Solution for Confirming Software Documentation Quality," vol. 7, no. 2, pp. 200–206, 2016.

[18] A. Jamal, R. Ali, A. Nouh, and F. Hossam, "Visualization analysis of feed forward neural network input contribution," *Sci. Res. Essays*, vol. 9, no. 14, pp. 645–651, 2014.

[19] M. Taheriyan, "Subject classification of research papers based on interrelationships analysis," in *Proceedings of the 2011 workshop on Knowledge discovery, modeling and simulation - KDMS '11*, 2011, p. 39.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[21] K. N. Vavliakis, A. L. Symeonidis, and P. A. Mitkas, "Event identification in web social media through named entity recognition and topic modeling," *Data Knowl. Eng.*, vol. 88, pp. 1–24, 2013.

[22] K. M. O. Nahar, M. Elshafei, W. G. Al-khatib, H. Al-muhtaseb, and M. M. Alghamdi, "Statistical Analysis of Arabic Phonemes for Continuous Arabic Speech Recognition," *Int. J. Comput. Inf. Technol.*, vol. 1, no. 2, pp. 49–61, 2012.

[23] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser, "Latent semantic analysis," in *Proceedings of the 16th international joint conference on Artificial intelligence*, 2004, pp. 1–14.

[24] D. M. Blei, "Probabilistic topic models,"

Commun. ACM, vol. 55, no. 4, pp. 77–84, 2012.

[25] H. Zhang and C. Wang, "A Robust Image Watermarking Scheme Based on SVD in the Spatial Domain," *Futur. Internet*, vol. 9, no. 3, p. 45, 2017.

[26] K. Loukhaoukha, "Image watermarking algorithm based on multiobjective ant colony optimization and singular value decomposition in wavelet domain," *J. Optim.*, vol. 2013, 2013.

[27] S. Gan, Y. Chen, S. Zu, S. Qu, and W. Zhong, "Structure-oriented singular value decomposition for random noise attenuation of seismic data," *J. Geophys. Eng.*, vol. 12, no. 2, pp. 262–272, 2015.

[28] N. M. Makbol and B. E. Khoo, "A new robust and secure digital image watermarking scheme based on the integer wavelet transform and singular value decomposition," *Digit. Signal Process. A Rev. J.*, vol. 33, pp. 134–147, 2014.