PAPERNERDS: A RESEARCH DIGITAL LIBRARY AND UNDERGRADUATE THESIS PORTAL WITH TEXT MINING ALGORITHMS

I. INTRODUCTION

Data grew exponentially every day. Almost all types of institutions and organizations stored data digitally. According to Sagayam (2012), this generally came as textual data in digital libraries and other repositories. In the College of Computer Studies (CCS) of Laguna State Polytechnic University Santa Cruz Campus, the Research Implementing Unit Head (RIUH) or the research coordinators also utilized digital storage to monitor the undergraduate thesis progress of their students. However, as their data grew larger, it became difficult for the research coordinators to track the undergraduate thesis progress of their students. Furthermore, it didn't allow the student researchers to publicly search for relevant documents related to their study through the Internet.

A public access to research papers and a platform for the undergraduate thesis process of their students was figured to overcome the aforementioned limitations. The integration of the said features into a web application would benefit not just those involved in the undergraduate thesis process, but also those researchers outside of the college.

The goals mentioned above were vastly enhanced with the integration of text mining algorithms, a technique for discovering patterns from collections of documents according to Gharehchopogh & Khalifelu (2011). With text mining, the stakeholders were given a better representation of information based on their requirements. Specifically, the integration of Information Retrieval (IR), a method for relevant research papers in a document collection according to a statistical similarity measure based on a user's query according to Hull & Grefenstette (1996), was implemented. Next, the integration of topic modeling, a technique for finding latent or hidden topics within document collections according to Anupriya & Karpagavalli (2015), recommended related research papers to users. Lastly, the integration of a clustering algorithm, a method for dividing collections of texts into their different clusters according to Yao, et al. (2012), helped suggest the most appropriate group and technical panel members for the students.

Thus, the study aimed to develop Papernerds, a digital library, and an undergraduate thesis portal, that can be used for searching relevant research papers, recommendation of related research papers, and suggestions of group and technical panel members through text mining algorithms for the undergraduate thesis process of students in the College of Computer Studies (CCS) at Laguna State Polytechnic University SCC.

II. RESEARCH OBJECTIVES

The study's goal was to design and develop "Papernerds: A Research Digital Library and Undergraduate Thesis Portal with Text Mining Algorithms" that stored research papers in a public digital library and dealt with some of the undergraduate thesis

processes with research stakeholders for the College of Computer Studies (CCS) in Laguna State Polytechnic University SCC with the use of topic modeling and information retrieval algorithms for the research digital library, and a clustering algorithm for the undergraduate thesis portal, and test the performance of the algorithms integrated in the web application.

Specifically, this study aimed:

1. To design and develop a web application that included a public digital library for research papers and a web portal for undergraduate theses for the College of Computer Studies (CCS) at Laguna State Polytechnic University SCC;
2. To test LDA topic modeling based on coherence and perplexity and K-means clustering based on homogeneity, completeness, and V-measure scores to identify which to integrate to the system for research papers recommendations and student group and technical panel member suggestions.
3. To examine the performance of the models developed integrated into the system through actual testing.

III. LITERATURE OF THE STUDY

Digital libraries had been around for a long time. Cabrerizo et al. (2015) stated that it had proven to be beneficial in providing information between students, researchers, etc., especially with the use of the World Wide Web (WWW) through the Internet. In fact, Watstein et al. (1999) stated that even the WWW was a digital library since it was a collection of web pages. It was essential to be developed, because for instance as stated by Marchionini (1995), digital libraries served as a sharing platform in institutions to serve books, periodicals, and even videos.

The implementation of a web portal was deemed necessary for the researchers' study since according to Eldrandaly (2009), it made it easier for multiple users to access information while connecting them to a single source of information, data, and services. Further, Mansourvar & Yasin (2010) stated that web portals had become increasingly crucial in institutions because of students' management of information online. For instance, Del Rosario et al. (2016) developed a thesis portal for their university, which covered the thesis process of different thesis documents.

Beyond system development, the researchers wanted to integrate the use of text mining with machine learning (ML) algorithms. Waegel (2006) defined text mining as the extraction of potentially useful information from textual data. Baharudin et al. (2010) stated that it had gained more importance as a result of the growth of electronic documents from various sources. Machine learning, as defined by Sah (2020) was the study of computer algorithms that provided systems the ability to automatically learn and improve from experience, and it was classified into primary categories: supervised, unsupervised, and reinforcement learning. Unsupervised learning, in which the data given were usually about the real world according to Celebi & Aydin (2016), was used by the researchers because supervised learning, according to Kotsiantis et al.

(2007), required a dataset with known labels and reinforcement learning, as stated by Bhatt (2018), learns by trial and error.

Several topic models were considered by the researchers from various studies. Rosario (2000) stated that LSI, the origin of topic models, was not an authentic topic model since it was not a probabilistic topic model. Probabilistic topic models include pLSA and LDA. However, Cai et al. (2008) stated that pLSA ran into overfitting which occurs when a model learned the detail and noise in the training data to the point that it would lower the model's performance on new data. Therefore, LDA, as defined by Blei et al. (2003), as a generative probabilistic topic model, which would the issues encountered with pLSA, was the chosen by the researchers.

There were two most commonly used similarity measures used for different tasks according to various studies. Lesot et al. (2009) pointed out that IR systems require similarity measures usually for search results ranking. Euclidean distance, as defined in the study by Huang (2008), was a similarity measure usually only used in clustering problems, where distance between two points was calculated. On the other hand, cosine similarity as stated by Rahutomo et al. (2012), was the metric usually used in IR systems, where the cosine value or angle between the term vectors of two documents was calculated. Therefore, cosine similarity was chosen.

Studies of different clustering algorithms reveal which was best for clustering tasks. Hierarchical Clustering, as pointed out by Xu & Wunsch (2005), represented results in a binary tree or dendrogram. On the other hand, K-means, the most popular clustering algorithm, was discussed by Wagstaff et al. (2001) as a method used to divide data into *k* groups. In comparison in terms of performance, Kaushik & Mathur (2014) pointed out that the performance of K-means was better than Hierarchical Clustering.

In evaluating the algorithms, several studies regarding the topic modeling, clustering, information retrieval, were discussed. For the LDA topic modeling, it was evaluated based on perplexity and topic coherence. Jacobi et al. (2016) defined perplexity as a metric for how well the model predicts a sample, with a lower perplexity indicating a better prediction. Topic coherence, on the other hand, was a metric that helped in distinguishing between topics that were semantically interpretable, and thus be able to identify the optimal number of *k* clusters.

Moreover, information retrieval was evaluated based on recall, precision, average precision (AP), and mean average precision (MAP). Sathya Bama et al. (2015) stated that in calculating recall, the number of relevant records retrieved was divided by the total number of relevant records. Further, in calculating precision, the number of relevant records retrieved was divided by the total number of irrelevant and relevant records retrieved. AP and MAP metrics were used to consider the ranking of the search results. Trotman (2005) defined AP for a single query as the sum of precisions divided by the number of relevant documents in the ranked list. On the other hand, Turpin & Scholer

(2006) defined MAP as the mean average scores over a set of queries.

Next, the K-means clustering algorithm was evaluated based on the silhouette coefficient method. Kumar (2020) stated that the method would determine the suitable number of *k* number of clusters since it was a measure of how similar an object is to its own cluster compared to other clusters.

In selecting the software development model, the SCRUM software methodology was found to be the most appropriate for the study of the researchers. This development model, as stated by Szalvay (2004), was a type of agile software development model, which was based on the idea of incremental and iterative development. The Waterfall model wasn't chosen even though it's relatively popular since according to Mishra & Dubey (2013), the development could not proceed to the next phase unless and until the previous phase was fully accomplished.

IV. METHODOLOGY

Research Design

The researchers of this study used descriptive research and developmental research design methods. As stated by Fireman Kramer (1985), descriptive research was regarded as the initial step in the development of new knowledge that described the characteristics of variables under the study that leads to a tentative hypothesis. This method was used by the researchers to gather the information that systematically described the study through interviews. The researchers, therefore, were able to thoroughly investigate the background of the research problem before further research was carried out.

On the other hand, developmental research, as defined by Richey (1994) as the systematic study of designing, developing, and evaluating instructional programs, processes, and products that must meet the criteria of internal consistency and effectiveness, was used by the researchers to determine the features to be developed for the system. Also, it was used to examine the consistency and efficacy of the development of the system.

Population of the Study

The target users of the study were composed of various stakeholders involved in the undergraduate thesis of the College of Computer Studies (CCS) department and other departments in Laguna State Polytechnic University SCC.

Specifically, the stakeholders of the study included the Research Implementing Unit Head (RIUH), the Dean/Associate Dean, the student researchers, the course instructors, the research advisers, and the subject specialists of the CCS. Additionally, the language critics, the technical editors, and the statisticians of the College of Arts and Sciences (CAS), the College of Teacher Education (CTE), the College of Engineering (COE), and the CSS, were also included. Lastly, the stakeholders of the study were also composed of guest users.

Data Model Generation

This section of data model generation enumerated the methodologies in generating

the data models of the algorithms starting with data preprocessing and then followed by evaluation of LDA and K-means, topic modeling, information retrieval, and clustering.

Data Preprocessing

Preprocessing of the text data was an essential step for text mining tasks. It allowed text to be transformed into a more digestible form so that machine learning algorithms could perform better. It helped to get rid of unhelpful parts of the data, or noise, through different techniques such as the conversion of all characters to lowercase, removal of stop words and punctuations, tokenization, stemming, lemmatization, and vectorization.

Lowercasing

Although often ignored, Tuncer (2020), stated that lowercasing text data was one of the simplest and most effective forms of text preprocessing. It could be used to solve most text mining problems, and it could be especially useful in significantly helping with the consistency of expected output.

Stop Words Removal

Stop words, according to Teha (2020), were words in English that didn't contribute much meaning to a statement, such as "a", "the", "is", and "are". They could be safely ignored without compromising the sentence's meaning.

Punctuation Removal

Punctuation removal was the removal of punctuations in a string. According to Ebrahim (2021), eliminating punctuation was a typical preprocessing step in many data analysis and machine learning tasks.

Stemming

Sawhney (2021) defined stemming as the process of reducing a word to its stem or root format. For instance, the terms, "branched," "branching," and "branches" could all be reduced to the same word "branch."

Tokenization

It is defined by Kannan et al. (2014) as the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens. The aim of tokenization was the exploration of the words in a sentence. The list of tokens would become the input for further processing such as parsing or text mining.

Lemmatization

According to Kotov (2017), lemmatization usually referred to doing things properly with the use of a vocabulary and morphological analysis of words, normally aimed to remove inflectional endings only and return the base or dictionary form of a word, which was known as the lemma.

Vectorization

In vectorization, according to Brownlee (2017), text data were encoded as integers or floating-point values for use as input to a machine learning algorithm.

System Development Methodology

The system was developed with the use of a Software Development Life Cycle (SDLC) model. According to Sree & Priya (2016), it was said that the Software Development Life Cycle was the most important aspect of software system development, and there were many various development methodologies used currently.

Out of all the software development methodologies, SCRUM was chosen by the researchers. As stated in the study by Srivastava et al. (2017), SCRUM was basically a lightweight, agile framework that outlines steps for managing and controlling the software and product development process. It was the combination of the Iterative model and the Incremental as builds were sequential and incremental in terms of the features to be developed with constant communication between the stakeholders. Moreover, SCRUM, according to the researcher, was created to speed up the development process. The SCRUM workflow consists of the product backlog, spring planning, sprint backlog, sprint with daily scrum, increment, sprint review, and sprint retrospective.

Product Backlog

The SCRUM workflow starts with the construction of a product backlog. The researchers first constructed a product backlog, which was simply a list of features or functionalities for the system or product.

The items on the list, which could be thought of as a "To-Do" list, were chosen based on suggestions from the Research Implementing Unit Head (RIUH) and ideas from the researchers. Also, besides listing the features, the researchers also prioritized the items. High-priority items were prioritized at the top of the list, while non-priority items were pushed to the bottom. Once the product backlog was built, it was regularly maintained by the researchers to make sure prioritization is correct and new items were added or removed.

Sprint Planning

In sprint planning, the researchers then addressed which product backlog items would be worked on. The researchers set up a meeting with one another in less than an hour to identify what could be delivered and how that could be achieved. Also, the duration of the sprint was addressed during the meeting. By the end of the sprint planning, the researchers were then ready to start working on the sprint backlog.

Sprint Backlog

The output from sprint planning was a sprint backlog. It was a list of product backlog items identified by the researchers for the sprint. For instance, the researchers included the first four (4) product backlog items in their first sprint, such as “search research papers”, “view related research papers”, etc.

Sprint

Following the creation of the sprint backlog, the researchers then began the sprint. Whenever a sprint begins, the researchers or the developer start their work. The developer built the features identified with the help of the sprint backlog with a limit to a set period of time during which the work had to be completed. During the sprint,

the researchers held a daily scrum, or a short daily standup meeting in less than 15 minutes and discussed the progress the researchers' developer was working on. Also, daily scrums were held by the researchers to ensure everything was on the right track.

Increment

The outcome of a sprint was a usable product called an increment or product increment. The increment was the integration of all the completed items done by the researchers or developer based on the sprint backlog. The researchers then marked the completed items as "Done" to show that it was in a usable condition which could be made available to the end-user in the form of a system.

Sprint Review

At the end of a sprint, the researchers and the Research Implementing Unit Head (RIUH), a key stakeholder, reviewed the outcome of a sprint in an event called sprint review. In sprint review, the researchers and the RIUH met to review the completed work and addressed any concerns.

Sprint Retrospective

After the sprint review, the researchers convened for another meeting in less than an hour to wrap up the sprint in a phase called sprint retrospective. In this phase, the researchers addressed what went well, what could be improved, and what they would be committed to improving in the next sprint. For the next sprint, the processes in the SCRUM workflow, starting with sprint planning, were repeated.

V. RESULTS AND DISCUSSION

RESEARCH OBJECTIVE 1: To design and develop a web application that included a public digital library for research papers and a web portal for undergraduate theses for the College of Computer Studies (CCS) at Laguna State Polytechnic University SCC.

To design and develop Papernerds, the researchers conducted interviews with the Research Implementing Unit Head (RIUH) in order to collect necessary information, such as research papers, and understand the undergraduate thesis process.

Based on the interviews, the researchers also consolidated a list of features to be developed, listed in a product backlog, which was the first step in the SCRUM software methodology. It was essentially a "To-Do" list, wherein high-priority items were prioritized at the top of the list, while non-priority items were pushed to the bottom. Sprint planning then took place on each iteration in order to identify the researchers would be working on in a series of sprints. After each sprint, a usable software called an increment, would have been developed. Sprint review and sprint retrospective were then undertaken to review and reflect on the process respectively.

The developed research digital library and undergraduate thesis portal were shown in the following:

The search bar and the "Sign In" button were shown in the figure above. The search bar allowed anyone to access the public research digital library of the web application, while the "Sign in" button allowed registered users to log in to the web application.

Upon clicking the “Sign in” button, the web application would show a login panel where users could log in with their username and password. Moreover, if a user forgot their password, they could click the text “Forgot your password?” Then, they would be redirected to another page to configure their forgotten password. Lastly, upon clicking the “Sign up” button, the user would be asked to enter their details to register.

The search feature of the application allowed users to retrieve a ranked search results of their input from the stored research papers in the database of the web application. The users would be able to distinguish a non-relevant research paper with this.

The page for the summary of a research paper was separated into two columns. The first column showed the abstract of the research paper, and the student researchers, technical panel, undergraduate program, year, number of views, and a button to view the full paper. The second column on the other hand simply showed the related research papers.

Upon scrolling the page for the summary of the research paper, the button to view the full paper as a 10-pager in PDF format could be seen. However, if a user wasn’t logged in, the user wouldn’t be able to perform the action.

The undergraduate thesis process is divided into four (4) stages namely, pre-proposal, proposal, research colloquium, and final oral defense. The web application works step by step in accomplishing all the steps for the undergraduate thesis process. An additional second column was added to represent the web portal of the web application.

For instance, the group selection step, under the pre-proposal stage, of the web application allowed users to get group member suggestions for their undergraduate thesis. It explained the details of each step under the “Details” section. Moreover, the deliverables for the step, where users could download and upload the documents, was indicated just right above the “Confirm” button.

RESEARCH OBJECTIVE 2: To test LDA topic modeling based on coherence and perplexity and K-means clustering based on homogeneity, completeness, and V-measure score to identify which to integrate to the system for research papers recommendations and student group and technical panel member suggestions.

The researchers conducted a series of tests based on the datasets prepared. The datasets, abstracts in the form of paragraphs and skills in the form of keywords of not more than five (5) words, were kept constant throughout the tests for fair comparison of LDA topic models and K-means clustering models.

A graph with k number of topics, ranging from 0 to 30, was plotted in order to visualize the perplexity scores of the LDA topic model within the range. For instance, the topic model on *k* number of topics of 30 gave a perplexity score of -6.720. The score mentioned indicated that the topic model

could very well predict a new sample, since a lower perplexity was better.

Moreover, a graph with *k* number of topics, ranging from 0 to 30, was plotted once again to visualize the coherence scores of the LDA topic model within the range. For instance, the topic model on *k* number of topics of thirty (30) gave the highest coherence score of 0.756. The high coherence score implied that the topics that would be returned by the topic model would be highly interpretable to humans.

Finally, a graph with *k* number of topics, ranging from 0 to 30, was plotted in order to visualize the perplexity scores of the topics LDA topic model within the range. For instance, the topic model on thirty (30) *k* number of topics gave a perplexity score of -5.141. The very low perplexity score result indicated that the topic model could very well predict a new sample better than that of the previous test on the abstracts dataset.

In comparison to the abstracts dataset, another graph with *k* number of topics, ranging from 0 to 30, was plotted in order to visualize the coherence scores of the LDA topic model within the range. The topic model on all k number of topics gave a constant coherence score of 0.698. The result gave an interpretation that the topic model on all number of topics have the same level of interpretation which indicates that the topics would all just be similar to each other.

Therefore, the researchers determined that LDA topic models worked perfectly with longer texts in determining the topics for distributions since the topic model on the abstracts dataset was found to be more coherent than that of shorter texts with unreliable interpretations. Further, although the perplexity score on the shorter texts gave a better measure for predicting a new sample, it wasn't critical since its coherence score was unreliable.

For the testing of the K-means clustering on the skills dataset, it received a perfect result of 1.000 on all the metrics. It was implied that the K-means clustering model was perfectly homogenous and complete. The clustering was perfectly homogeneous since in each cluster the data points were of the same class label, and it was perfectly complete since all data points belonging to the same class were clustered into the same cluster. Thus, their harmonic mean, measured using a V-measure score, was also 1.00.

Lastly, for the testing of the K-means clustering on the abstracts dataset, a totally different result was received. The low homogeneity score of 0.030 indicated that the cluster contained several data points that were members of multiple classes. Moreover, the completeness score of 0.093 indicated that a few data points belonging to the same class were clustered into the same cluster. Thus, the clustering's V-measure score was 0.045, indicating a poor cluster.

The researchers, therefore, concluded that the K-means clustering algorithm worked well with the short texts from the skills dataset as opposed to longer texts from the abstracts dataset.

RESEARCH OBJECTIVE 3: To examine the performance of the models developed integrated into the system through actual testing.

The researchers conducted a series of tests in order to examine the performance of the topic model integrated into the system. In order to carry out the test, the researchers prepared three (3) word intrusion and three (3) topic intrusion tasks presented to one of their research members.

The subject was tasked to identify the intruder word from the sets. It was shown in the table that the subject never got an incorrect prediction. Therefore, it was concluded that the LDA topic model integrated into the system was said to have an excellent coherence of words within a topic since the subject got all predicted answers right based on the three (3) sets of word intrusion tasks.

Afterwards, the subject was tasked to identify the intruder topic, represented by the eight highest-probability words within that topic, from the sets. The results show that the subject got one (1) incorrect prediction. In conclusion, the LDA topic model integrated into the system is said to have a good coherence of topics, since based on the three (3) sets of topic intrusion tasks, the subject got two (2) correctly predicted answers out of 3.

The web application developed by the researchers integrated two (2) K-means clustering models for suggestions. The performances of the models were tested on ten (10) datasets for each.

The K-means model of the system for student group member suggestions got one (1) incorrect prediction out of ten (10), yielding an accuracy score of 0.90. On the other hand, the K-means model for technical panel member suggestions got two (2) incorrect predictions with 8 true predictions, yielding an accuracy score of 0.80.

The researchers, therefore, concluded that the K-means clustering models integrated into the web application were highly accurate since they had very few false predictions.

The researchers conducted tests on the information retrieval system of the web application using the three (3) query sets prepared in order to examine its performance. To quantify its performance, recall and precision were calculated on each of the query sets, and their average precision (AP), and mean average precision (MAP) were calculated afterward.

The ranked information retrieval system of the web application retrieved a total of 3 documents from the first query set, “SMS notification,” ranked from the highest similarity to the lowest. The calculated recalls were 0.33, 0.66, and 1.00, while the calculated precisions were all 1.00.

The ranked information retrieval system of the web application retrieved a total of two (2) documents from the second query set, “barangay profiling,” ranked from the highest similarity to the lowest. The calculated recalls were all 1.00, while the calculated precisions were 1.00 and 0.50.

The ranked information retrieval system of the web application retrieved a

total of three (3) documents from the second query set, “municipality,” ranked from the highest similarity to the lowest. The calculated recalls were 0.17, 0.33, and 0.50, while the calculated precisions were all 1.00.

The result of the performance of the ranked information retrieval system was calculated using the mean average precisions (MAP) from the calculated average precisions (AP) of the query sets prepared by the researchers. It showed that the mean average precision (MAP) is 0.92 or 92%.

VI. CONCLUSION

Based on the findings of the study from the objectives, the researchers have reached the following conclusions and generalizations:

1. In developing the web application, the researchers noticed that the development was fast and efficient using the Python programming language since it was used to create the machine learning algorithms and develop the other back-end functionalities, instead of using other programming languages for multiple tasks.
2. The researchers figured that topic models worked particularly well with longer texts as opposed to clustering algorithms which worked better with shorter texts.
3. After performing the performance evaluation of the models or algorithms, the researchers found the that performance testing conducted on the LDA topic model using word and topic intrusion tasks was effective.

VII. RECOMMENDATIONS

Based on the conclusions, the researchers have reached the following recommendations:

1. The researchers recommended the use of the Python programming language when building a web application integrated with machine learning algorithms for faster development.
2. Based on the findings, the researchers recommended the use of K-means for clustering short texts and LDA topic modeling only for long texts for correct clustering and topic modeling respectively.
3. The researchers recommended the use of the word intrusion and topic intrusion tasks for evaluating the performance of the LDA topic models since they were effective in capturing the coherence of words and topics.