International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I2P110

Volume 72 Issue 2, 82-91, February 2024
© 2024 Seventh Sense Research Group®

Original Article

Tool for Oversight Syllabuses and Feedback

Ritu Sodhi[1], Jitendra Choudhary[2], Ruby Bhatt[3]

[1]*Computer Applications, Medi-Caps University, Indore, Madhya Pradesh, India.*
[2,3]*Computer Science, Medi-Caps University, Indore, Madhya Pradesh, India.*

[1]*Corresponding Author: ritu.sodhi@medicaps.ac.in*

Received: 03 October 2023 Revised: 14 December 2023 Accepted: 26 December 2023 Published: 03 February 2024

Abstract *- The creation of curricula and syllabi is a crucial component of the educational system. The process of developing curricula involves industries as well. The study has been done to oversee syllabuses. Work has been done in the area of sentence-similar tests. The prior study does not compare the syllabuses of the same courses, nor does it offer the designer of the syllabus advice for the topics that might be included in it. This paper suggests a mechanism for developing and maintaining the syllabus. Also, it will consider industry feedback. This tool compares feedback from various industries after receiving input from them and makes recommendations for subjects and contents that syllabus creators might want to include in their curriculum. This study used the streamlit framework and spacy for semantic comparison. This study will create a tool to oversee the syllabus. Universities can solicit input from industries using this technology on two different levels. Initially, it asks for advice on courses the institution can include in the curriculum. Second, it can accept comments regarding the subject matter of any course. Industry can advise the university on the newest/upcoming innovations. After receiving feedback from various industries, this model compares the feedback and provides the curriculum designer with recommendations for topics that might be covered in the syllabus. In this study, the actual industry feedback from 10 industry experts has been taken and semantically compared their feedback on contents, calculated the weightage of topics, and displayed the topics in decreasing order of weightage. Conclusion: The model for creating a curriculum and putting it into practice will be provided by this research, allowing us to follow a procedure. It bridges the gap between academia and industry.*

Keywords *- Curriculum, Comparison, Feedback, Industry, Syllabus, University.*

1. Introduction

Overview of topic: This paper proposed a tool for syllabus creation and reducing the gap between industries and universities. Universities create new computer science syllabuses and update the existing ones as and when needed. The need to introduce new courses and update the existing ones is due to changing industry needs and the introduction of new university programs.

1.1. Research Gap

The study has been done to create a syllabus management tool. Studies also have been done on text similarity techniques. However, text similarity methods are not used to compare the feedback given by the industries to give suggestions to the faculties.

1.2. Problem Statement

Research is done on creating a single platform to create, update, and delete the syllabus. Research has been done on taking feedback from industries. However, they did not give a platform to compare the feedback and give suggestions about the courses and topics to be included in the curriculum. There is a need for a systematic model for overseeing syllabuses and feedback.

1.3. The Rationale of the Paper

This study provides the syllabus management tool. It also provides a tool to reduce the gap between industries and academia by taking the feedback from industries and suggesting the topics to the faculties after comparing the feedback.

1.4. Methodology

In this paper, a tool for managing the curriculum was proposed. Our tool can be used by those who build and update syllabuses. Our solution also enables receiving industry-specific input. Input will be sought regarding (1) upcoming new courses and (2) the syllabus content. Universities consider comments before deciding whether or not to add new courses to their curriculum. Our tool compares suggested content by industries using semantic analysis after receiving feedback (regarding the syllabus contents). The tool used spacy for semantic comparison. After comparison, our tool suggests them to the creator of the curriculum.

1.5. Research Statement

The suggested tool offers a solitary platform.

1) For the creation of the syllabus and updating of the syllabus.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

2) To take industry input.
3) To compare different feedback provided by industries, compare them semantically and determine the weight of each issue.
4) To make recommendations to the syllabus designer regarding topics that can be included in the syllabus based on the weighted average.

1.6. Outline

This paper is organized as follows: In Section 2, this research discusses the background and related work. Section 3 presents the prototype of the governing syllabus and feedback. Section 4 contains the pseudocode and output of the semantic comparison of feedback from industries. Section 5 presents data collection and result analysis. Section 6 concludes our research work.

Table 1. Related work on developing a repository of syllabuses and tools for managing syllabuses

Sl. No	Year of Study	Findings	Comments
1.	2003	A database of syllabi was created by Matsunaga et al. [1]. An effective crawling system was employed in the article for these objectives.	Compiling a syllabi database but not comparing syllabi for the same course.
2.	2007	A repository for automated syllabi was made by Yu et al. [2]. A digital library for computer science courses was produced by the study. Because irrelevant links can appear in search results, finding a syllabus online can be difficult. Moreover, the online syllabus is not organized. A collection of syllabuses with quick and easy access could have a big impact on education. In this study, unstructured syllabi were transformed into structured syllabi using information recognition, segmentation, and classification. They developed a structured library for syllabuses that can be searched for textbooks, syllabus contents, and other syllabus-related information.	A digital library for computer science courses was produced by this study.
3.	2007	A repository, of course, syllabi for courses at US universities was developed by Tungare et al. [3]. A platform that teachers can use to design and publish syllabuses was also developed as part of this study.	This article offered a tool for creating syllabuses.
4.	2008	Abdous and he [4] developed a framework to make the task of creating a curriculum easier. The system enables the creation, modification, deletion, and sharing of syllabuses.	The study served as the foundation for developing a syllabus.
5.	2009	In Ireland, Joorabchi and Mahdi [5] sought to establish a national syllabus repository. A prototype for the syllabus repository was created and put to the test in this study. It is a framework for the national repository system that is semi-automated.	A syllabus repository and a semi-automated system were created by the study.
6.	2013	Because of numerous unrelated connections, searching the computer science curriculum on search engines does not work properly. Several machine learning classifiers were trained by Rathod and Cassel [6] to identify computer science curricula from search results.	This work developed classifiers to categorize computer science curricula using search engine results.
7.	2015	An application for the automation of syllabus development was created by Hussein et al. [7]. To guarantee that all course outcomes must be in line with program outcomes, the application also maps course outcomes and program outcomes.	Their research helps the faculties to align and map the course outcomes of a course with the program outcomes.
8.	2016	Using ontology to convert the unstructured syllabus into a structured syllabus and store it in a syllabus repository, Chung and Kim [8] develop a syllabus classification scheme.	This research used ontology to clean the syllabuses available on the web.
9.	2018	Guberovic et al. [9] developed a web application named CSyllabus. In this web application, the study creates the syllabus repository of eleven universities. This syllabus repository helps students and teachers access and compare various syllabuses. The study also implemented a comparison algorithm using Latent Semantic Indexing from the Gensim library. This study has taken feedback about their web application.	This study developed a repository for the syllabus. It also helps to access and compare various courses.
10.	2020	Agent-based modeling is used by Mosharraf et al. [10]; in this method, the syllabus repository is first built. The pertinent data is extracted after the syllabus repository has been created. The agent will ultimately decide what will be included in the syllabus.	The development of a syllabus repository was the focus of the study.

2. Background and Related Work

2.1. Establishment of a Repository for Syllabi and Tools for Maintenance of Syllabi

The research was conducted to create a repository for syllabuses and to extract the syllabuses from the repository. Because many web contents are irrelevant as per need and the available curricula on search engines are in an unstructured manner, data cleaning and extraction of important data are necessary. Research has been conducted to develop tools for managing syllabi. The studies on building repositories, text retrieval, transforming unstructured content into structured content, and producing content and publishing it to the web are all displayed in Table 1. The feasibility of developing a repository and framework for syllabi has been investigated. Any framework, however, does not advise the syllabus developer to incorporate the syllabus material.

2.2. Methods for Comparing Texts

Sentence-related tests are the subject of a great deal of research. It is a natural language processing component with several uses, including text retrieval, topic discovery, plagiarism detection, comparing DNA patterns, event detection, etc. Nevertheless, it has not been utilized to compare various feedback from the industry. Table 2 discusses studies on the necessity for text analysis, text analysis techniques, how to improve text analysis techniques already in use, and how to combine various text analysis techniques with various data sets.

The research discussed above focuses on similarity metrics across multiple datasets. However, a subject for further study is how they carry out the comparison of datasets relating to computer science keywords.

Table 2. Related work on text similarity techniques

Sl. No	Year of Study	Findings	Comments
1.	2005	On semantic similarity, Corley and Mihalcea [11] worked. In this study, the authors combine two approaches—word-to-word and text-to-text—and present a brand-new approach. This new approach outperforms more established semantic similarity approaches.	Better outcomes are obtained when two distinct approaches to semantic similarity detection are combined.
2.	2008	Evaluation of sentence similarity is important in various contexts, including text mining, question-answering, and text summarising. Fourteen techniques for determining the semantic similarity of texts were covered by Achananuparp et al. [12]. This study also addressed which approach works best with different types of data sets.	The article reviewed various techniques for comparing text semantic similarity and recommended the technique that works best with each kind of data set.
3.	2009	Pesquita et al. [13] reviewed semantic similarity methods used in biomedical ontologies and proposed their classification as node-based versus edge-based and pairwise versus group-wise. This article also did comparison research and examined the effects of their results.	The biomedical terminologies were examined and analyzed for semantic similarity, and comparative studies were conducted.
4.	2013	Several techniques are used by Rus et al. [14] to compare the semantic similarity of the two texts. Additionally, this article provides tools for expert manual annotation of semantic similarity using a SEMantic simILarity Annotation Tool.	They provided manual semantic annotation using a tool in addition to implementing several methods for semantic similarity.
5.	2013	Gomaa and Fahmy [15] talked about different text similarity metrics. Lexically or semantically related texts may be found. Using string-based techniques, the lexical similarity across texts can be compared. Corpus-based and knowledge-based (WordNet) methods can be used to identify semantic similarities between texts. This study went into great detail about string-based, corpus-based, and knowledge-based techniques.	This paper discussed many methods for comparing texts.
6.	2013	The approaches for determining semantic similarity were addressed by Slimani [16]. Also, this study assesses multiple semantic similarity approach categories. This study evaluates all of these semantic similarity measures to help researchers and practitioners choose the metric that best meets their needs.	This study reviewed and assessed a variety of semantic similarity approaches that aid in choosing the best-fit metric for the task at hand.
7.	2016	The strategies for identifying word and sentence similarities are contrasted by Atoum et al. [17]. Three approaches—corpus-based, knowledge-based, and hybrid approaches—are compared in this study. They discovered through this comparison that hybrid strategies outperform corpus-based and knowledge-based approaches in terms of results.	The study contrasted several sentence similarity calculations.

8.	2016	To model and control curricula semantically, teaching and learning methods need to be improved [8].	This study suggests that modeling and regulating the curricula on a semantic level are necessary.
9.	2018	A proposal to use text mining and semantic technologies to compare academic content was addressed by Saquicela et al. [18].	This study used natural language processing to examine similarities in academic content.
10.	2018	To determine the effectiveness of sentence similarity, Chen et al. [19] looked at the biomedical area. To assess the efficiency of sentence similarity criteria for sentence ranking, a study of PubMed documents was carried out. Neither lexical nor semantic measurements yield the expected results for sentence ranking.	The study used language similarity techniques in the field of biomedicine.
11.	2019	Quan et al. [20] integrate the attention weight approach with syntactic and semantic data to produce the ACVT kernel, a novel tree kernel for sentence similarity.	An ACVT kernel for sentence similarity was constructed in this study.
12.	2020	The results of text retrieval are required in many applications, making natural language one of the key approaches in artificial intelligence [21, 22].	There are various uses for natural language processing.
13.	2020	Jelodar et al. [23] assert that looking up the definitions of words in the area is necessary.	The semantic similarity was covered in the article.
14.	2020	Semantic analysis is carried out using the Word2vec approach [24].	Word2vec semantic analysis was employed by the researchers.
15.	2020	An Enhanced-RCNN model for sentence similarity is put out by Peng et al. [25]. The Enhanced-RCNN model's architecture is simpler than the BERT model's. The Enhanced-RCNN model outperforms baseline levels, according to the experimental data. Two data sets from the real world yield a competitive return for the model.	An enhanced RCNN model for sentence similarity was proposed in this study.
16.	2020	The state of similarity measurement research is carefully examined by Wang and Dong [26], who also weigh the advantages and disadvantages of current methods and develop a more thorough categorization description system for text similarity measurement algorithms. They also outline the direction of future research.	The research looks at the applications and drawbacks of string-based approaches, corpus-based methods, semantic text-matching methods, and graph-structure methods.
17.	2021	Advanced semantic analysis techniques provide a good level of accuracy, between 90 and 95 percent, according to Subhi et al. [27], who compared various methodologies and tools utilized for the task.	This study reviewed various tools and methodologies for performing semantic analyses and recommended the best one.
18.	2021	In their study of the most-read news articles on news websites, Singh and Singh [28] compare how two identical news articles—one in Hindi and one in English—refer to the same incident change in each language. A highlighted headline and link extractor has been used to pull the most important news stories for both Hindi and English from Google's news feed. The study first compared the Hindi news articles against English news stories by using Google Translator to translate them from Hindi to English. Second, they used cosine similarity, Jaccard similarity, and Euclidean distance measurements to calculate the news similarity score. The frequency of nouns and the following words are also found in the news items.	This methodology analyses the similarities between news stories and can identify the most popular news pieces.

3. Prototype of Governing of Syllabus and Feedback

Figure 1 depicts the design for the curriculum and feedback system. The login interface is specified by the "Login". The interface where a new syllabus can be created is designated as "New Syllabus." The interface for updating the current syllabus is displayed under "Update Syllabus." The interface that enables the faculties to view the industry feedback is described under "View Industry Input." The industry can provide comments about courses and contents by using the "Give Feedback About Courses and Contents" button. The data is processed using the term "processing." The "Feedback Analysis" is used to examine the comments made by professionals in the field. Figure 1 provides an overview of handling the syllabus and feedback. There are two different kinds of users: one is an industry specialist, and the other is a designer of curricula. The syllabus builder requires login credentials to create new syllabuses, update old ones, and view industry input. Two degrees of comment are possible from an industry expert. In the first place, regarding the courses that can be taught in universities, these courses can also include new technology. Second, about the details of the specific syllabus. However, no login information or password is required to provide comments on the courses and their contents.

A prototype of the tool is presented in Figure 1. The syllabus information and feedback are kept in a MySQL database. The syllabus's content is kept in the text file. A screenshot of the output of the implementation of the prototype of Figure 1 is given in Figure 2. After logging in, the syllabus creator can create a new syllabus. Update the syllabus and View the results of the comparison between feedback. Industries can give feedback without login.

The methodology followed for the syllabus construction process is firstly to create and implement an algorithm for creating, updating, deleting, and saving the course contents. Take inputs from the industry about courses and contents. Then, create and implement an algorithm to analyze the inputs from industries, calculate the weightage of various topics, and give suggestions to the syllabus creator according to the weightage calculated.

4. Comparison Between Feedback

Users can view the outcomes of the semantic comparison of industry feedback. Pseudocode 1 is shown below, and Figure 3 shows a screenshot of the results. The pseudocode compares the feedback semantically and calculates the weightage of each topic, with the topic with more weightage appearing before the topic with less weightage.

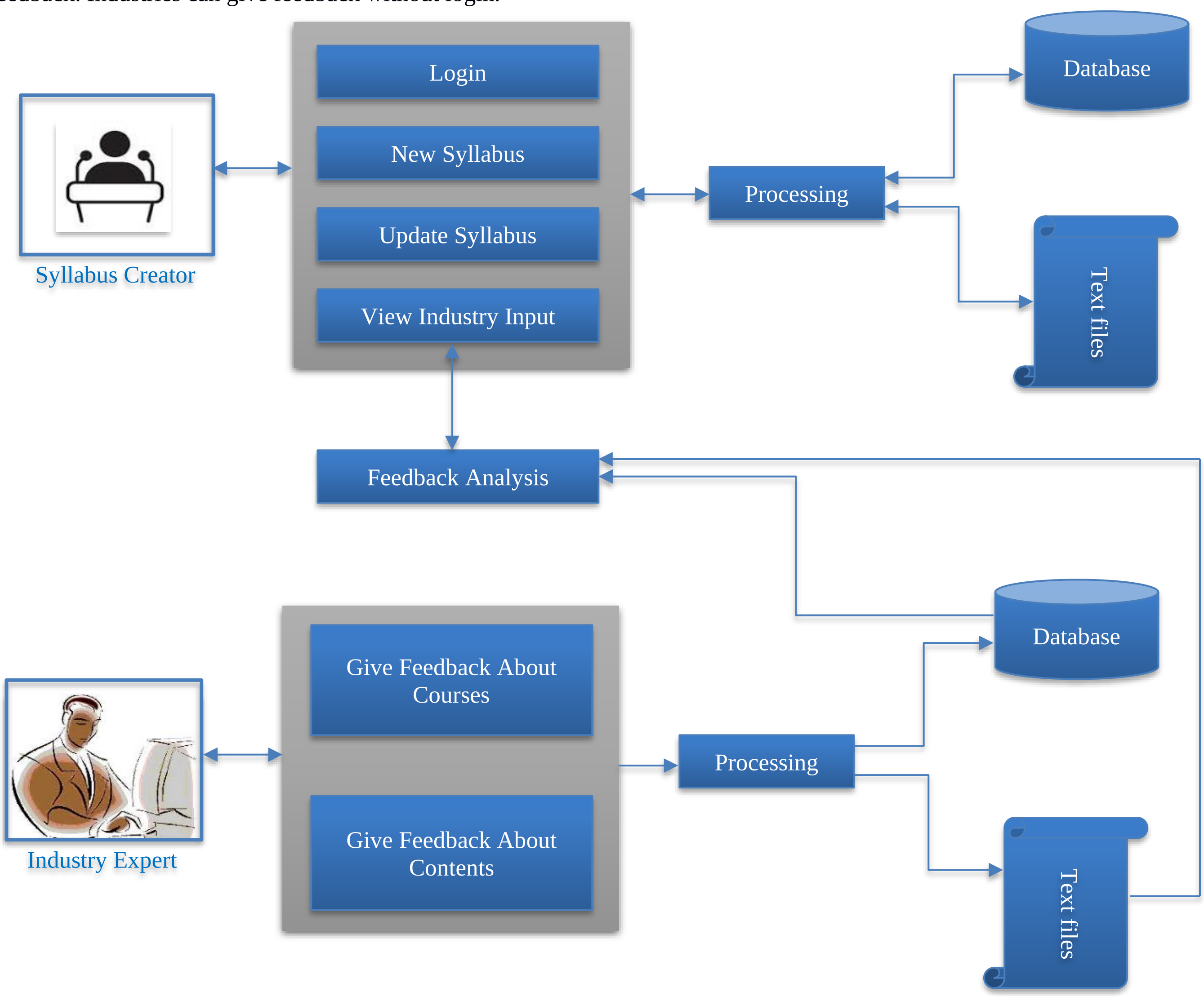

Fig. 1 Overview of handling of syllabus and feedback

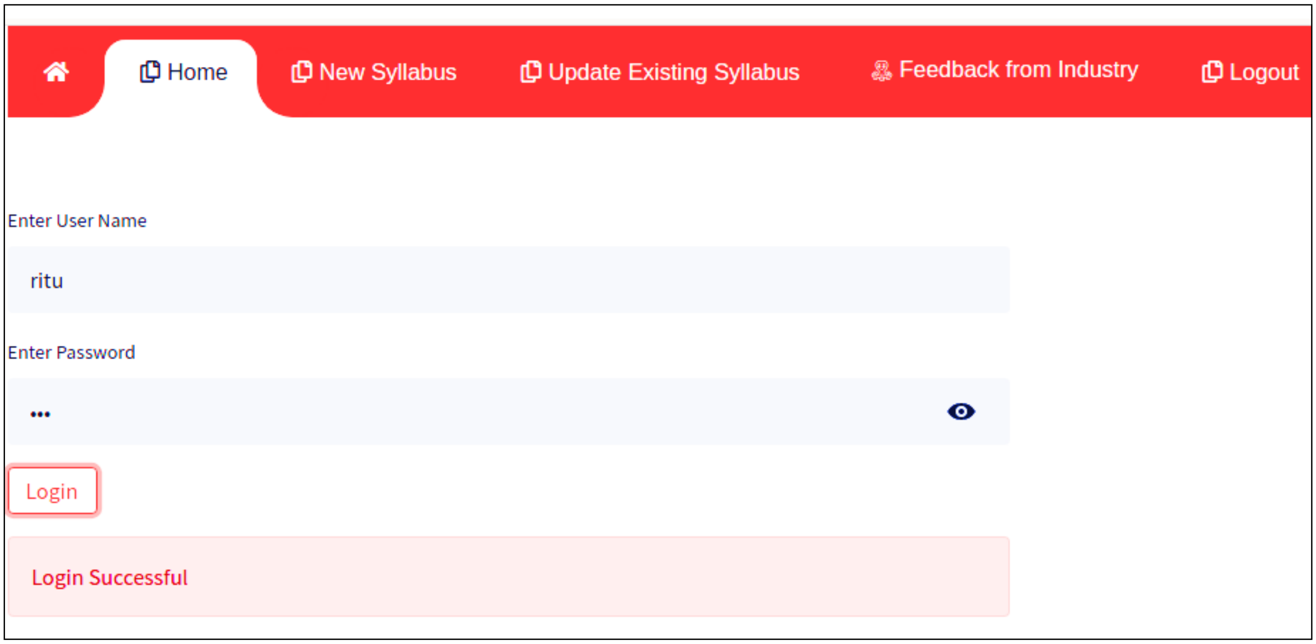

Fig. 2 Screenshot of the user login page

Pseudocode 1: Semantic comparison among industry feedback

```
        Read from the company table the total number of feedback that has been submitted for a specific course and record it
in n
        for i :=  1 to n do{
          Read the subsequent feedback from the company table;
          Separate the feedback about Punctuation Marks;
          Store the split feedback in different columns in the matrix m in row_i;
          The length of row_i of matrix m is stored in a[i];  //a[i] stores the number of topics in Feedback_i
        }
        i1:=0;
        for i:= 1 to n do    // Run the loop for the total  number of feedbacks
        {
           for j:=1 to a[i]  do     //Run the loop for total number of topics a[i] of i^th feedback (Feedback_i have a[i] topics)
           {
                for k:=1 to n do such that (i≠k)     // Run the loop from 1 to  n
           {
            for l:= 1 to a[k] do   //Run the loop from 1 to a[k] (Feedback_k have a[k] topics)
             {
               count[i,j] := coun[i,j] +semantic_simlarity(m[i,j],m[k,l]);
               //Find the topic m[i,j] weightage by comparing it to the topics of all the other feedback, then save
               //the results in count[i,j].
             }
           }
                     i1=i1+1;
                    C[i1] = count[i,j];     // Copy the weightage from count[i,j]  to C[i1]
                    s[i1] = m[i,j];      // Copy the topic from  m[i,j] to  s[i1]
            }
        }
        for j1:= 1 to i1-1 do       // order the topics by the sum of their overall semantic weights in descending order.
        {
           for m1 = j1+1 to i1 do
           {
            if(C[j1]<C[m1]) then
```

```
        {
         temp1:=C[j1];
         C[j1]:=C[m1];
         C[m1]:=temp1;
         temp2:=s[j1];
         s[j1]:=s[m1];
        s[m1]:=temp2;
         }
     }
}
r :=[];    // set up the array r
k2:=0;
for each w in s do
{
   if the w is not in r  then     // all of the feedback's duplicate topics should be removed.
  {
     k2 = k2+1;
    r[k2] = w;
  }
 }
print(r);  // Print all topics in descending weight order.
```

Below are three industry feedback samples for the contents of the C syllabus, which is the input to pseudocode 1. Company A [Company Number 0] given feedback: Function, Array, Pointer. Company B [Company Number 1] given feedback: Module, Array, Function, Pointer.

Company C [Company Number 2] given feedback: Module, Array, Arithmetic operators, Function. Table 3 gives some results after a semantic comparison between every two topics of two different industries after the execution of pseudocode 1.

This tool sorts the weightage of the topic “a” [Column 8 of the above table] in decreasing order and removes the duplicate topic [Column 3]. The topics having more semantic similarity come before the topics having less semantic similarity [Column 3]. The final output after semantically comparing feedback given by the industry is shown in Figure 3. After clicking on the “view industry inputs” tab, the syllabus creator can view these suggestions given by the industry to create a syllabus. As in Figure 3, the result is Function, Array, Pointer, Module, and Arithmetic Operators. This output matches the on-paper calculation.

Table 3. Results of semantic comparison of feedback given by industries

Company Number “i”	Feedback Number Give by Company “i”	Topic “a” Suggested by Company “i”	Company Number “j”	Feedback Number Give by Company “j”	Topic “b” Suggested by Company “j”	Semantic Similarity between Feedback “a” and Feedback “b”	Weightage of the topic “a”
0	0	Function	1	0	Module	0.5104954	0.510495398
0	0	Function	1	1	Array	0.4763164	0.986811766
0	0	Function	1	2	Function	1	1.986811766
0	0	Function	1	3	Pointer	0.4510327	2.437844468
0	0	Function	2	0	Module	0.5104954	2.948339866
0	0	Function	2	1	Array	0.4763164	3.424656234
0	0	Function	2	2	Arithmetic operators	0.4630707	3.887726927
0	0	Function	2	3	Function	1	4.887726927

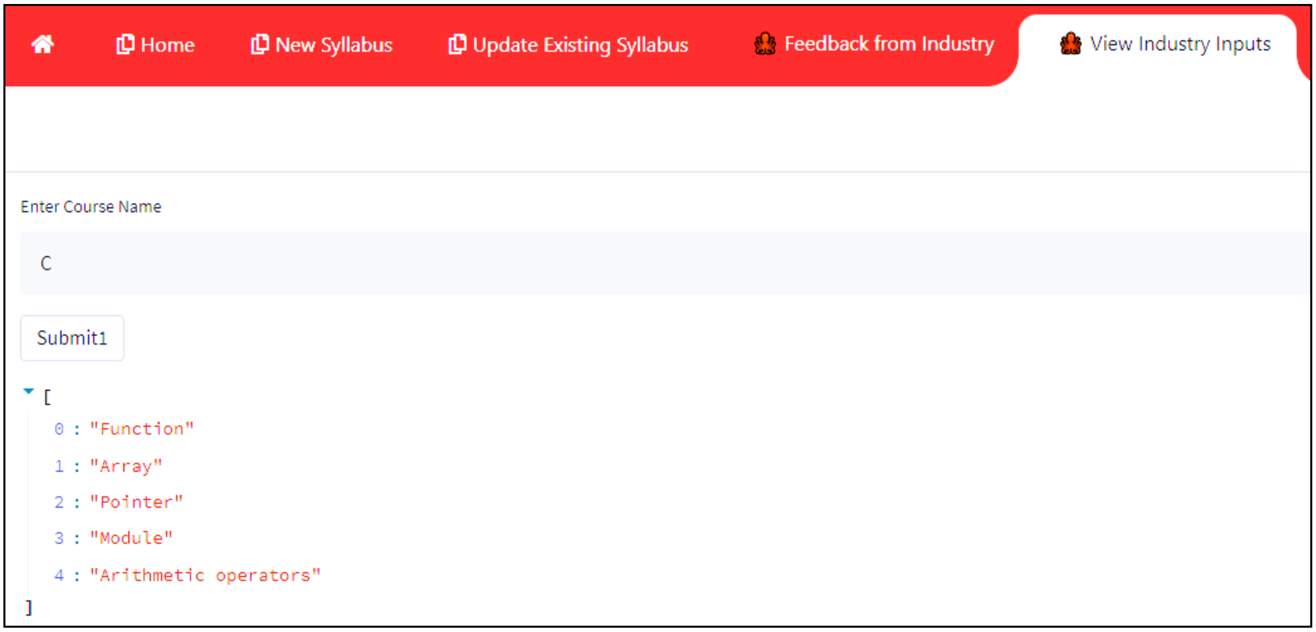

Fig. 3 Screenshot of the result of semantic comparison of feedback from industries

5. Data Collection and Result Analysis

Data was collected from the 10 Industry employees for taking feedback about the topics of "Core Java." The result of execution after semantic comparison among the feedback given to the industry is shown in Figure 3. Results are accurate when compared to manual results.

The results of the study are

- Designed and implemented the algorithm for handling the syllabus.
- Designed and implemented the model to take industry feedback.
- Designed and implemented the algorithm to analyze the industry feedback using text similarity methods and suggest the topics to the syllabus creator.

The tool allows syllabus creators to create new syllabuses and update the existing ones. The tool helps the universities to take feedback from industries.

Industries can give feedback related to what courses universities can add to their curriculum. Industries can also give feedback regarding the content that can be added to a particular course. This paper also contains pseudocode for semantic analysis of feedback given by industries. After semantic analysis, it will give suggestions to the syllabus creator about the content that can be added to their syllabus by the syllabus creator. Data from 10 industry experts are also taken and compared semantically.

6. Conclusion

6.1. Research Statement

This paper proposed and implemented a Tool for Oversight Syllabuses and Feedback.

6.2. Main Points

In this paper, a tool for creating syllabuses was suggested. Using this technology, syllabus writers can update and develop the new syllabus. This tool will assist universities in receiving input from industries. Industries can provide input on courses that universities should include in their curriculum. The contents that can be added to a given course can also be suggested by industries.

Our approach consists of a semantic analysis of industry feedback. The tool conducted semantic analysis on a range of industry inputs. After doing semantic analysis, our tool will make recommendations to the syllabus designer regarding the contents they can include in their syllabus.

6.3. Significance

By making the anticipated skill set and associated gaps in the academic curriculum more widely known, this paradigm will be advantageous to both the software business and academia. This study contends that greater industry-academia collaboration is vital to closing the skills gap. Doing so requires putting light on the key competencies in the software industry and understanding how they are addressed in the curricula.

The process should be such that it reduces the gap between the software industry and academia and reduces the manual process of syllabus construction by providing a syllabus management tool. As requirements from industries change frequently, automating the syllabus creation process is necessary. In the future, this work can be extended to automatically create syllabi by using syllabuses industry feedback.

References

[1] Yoshihiro Matsunaga et al., "A Web Syllabus Crawler and its Efficiency Evaluation," *Proceedings of the International Symposium on Information Science and Electrical Engineering*, Fukuoka, Japan, pp. 565-568, 2003. [Google Scholar] [Publisher Link]

[2] Xiaoyan Yu et al., "Using Automatic Metadata Extraction to Build a Structured Syllabus Repository," *International Conference on Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pp. 337-346, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[3] Manas Tungare et al., "Towards a Syllabus Repository for Computer Science Courses," *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, pp. 55-59, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[4] M'hammed Abdous, and Wu He, "A Design Framework for Syllabus Generator," *Journal of Interactive Learning Research*, vol. 19, no. 4, pp. 541-550, 2008. [Google Scholar] [Publisher Link]

[5] Arash Joorabchi, and Abdulhussain E. Mahdi, "An Automated Syllabus Digital Library System for Higher Education in Ireland," *The Electronic Library*, vol. 27, no. 4, pp. 640-658, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[6] Nakul Rathod, and Lillian Cassel, "Building a Search Engine for Computer Science Course Syllabi," *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 77-86, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bassam Hussein et al., "A Web-Based University Courses Syllabi Generator," *Computer Engineering and Intelligent Systems*, vol. 6, no. 11, pp. 1-7, 2015. [Google Scholar] [Publisher Link]

[8] Hyunsook Chung, and Jeongmin Kim, "An Ontological Approach for Semantic Modeling of Curriculum and Syllabus in Higher Education," *International Journal of Information and Education Technology*, vol. 6, no. 5, pp. 365-369, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Emanuel Guberovic et al., "In Search of a Syllabus: Comparing Computer Science Courses," *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, pp. 588-592, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Maedeh Mosharraf, and Fattaneh Taghiyareh, "Automatic Syllabus-Oriented Remixing of Open Educational Resources Using Agent-Based Modeling," *IEEE Transactions on Learning Technologies*, vol. 13, no. 2, pp. 297-311, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Courtney Corley, and Rada Mihalcea, "Measuring the Semantic Similarity of Texts," *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pp. 13-18, 2005. [Google Scholar] [Publisher Link]

[12] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen, "The Evaluation of Sentence Similarity Measures," *International Conference on Data Warehousing and Knowledge Discovery*, Berlin, Heidelberg, pp. 305-316, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[13] Catia Pesquita et al., "Semantic Similarity in Biomedical Ontologies," *PLOS Computational Biology*, vol. 5, no. 7, pp.1-12, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[14] Vasile Rus et al., "Semilar: A Semantic Similarity Toolkit," *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, pp. 63-168, 2013. [Google Scholar] [Publisher Link]

[15] Wael H. Gomaa, and Aly A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13-18, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[16] Thabet Slimani, "Description and Evaluation of Semantic Similarity Measures Approaches," *International Journal of Computer Applications*, vol. 80, no. 10, pp. 25-33, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[17] Issa Atoum, Ahmed Otoom, and Narayanan Kulathuramaiyer, "A Comprehensive Comparative Study of Word and Sentence Similarity Measures," *International Journal of Computer Applications*, vol. 135, no. 1, pp. 10-17, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Victor Saquicela et al., "Similarity Detection among Academic Contents through Semantic Technologies and Text Mining," *IWSW 2018 - International Workshop on Semantic Web*, pp.1-12, 2018. [Google Scholar] [Publisher Link]

[19] Qingyu Chen et al., "Sentence Similarity Measures Revisited: Ranking Sentences in PubMed Documents," *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, Washington, DC, USA, pp. 531-532, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zhe Quan et al., "An Efficient Framework for Sentence Similarity Modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 853-865, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Xiaofang Liao, and Zijiang Zhu, "Classification of Natural Language Semantic Relations Under Deep Learning," *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, Dalian, China, pp. 1025-1027, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Artem A. Maksutov et al., "Knowledge Base Collecting Using Natural Language Processing Algorithms," *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, St. Petersburg and Moscow, Russia, pp. 405-407, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] Hamed Jelodar et al., "A Collaborative Framework Based for Semantic Patients-Behavior Analysis and Highlight Topics Discovery of Alcoholic Beverages in Online Healthcare Forums," *Journal of Medical Systems*, vol. 44, no. 101, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Xiaolong Wang, Xingtong Dong, and Shuxin Chen, "Text Duplicated-Checking Algorithm Implementation Based on Natural Language Semantic Analysis," *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, pp. 732-735, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shuang Peng et al., "Enhanced-RCNN: An Efficient Method for Learning Sentence Similarity," *Proceedings of the Web Conference 2020*, Taipei, Taiwan, pp. 2500-2506, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[26] Jiapeng Wang, and Yihong Dong, "Measurement of Text Similarity: A Survey," *Information*, vol. 11, no. 9, pp. 1-17, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[27] Dastan Hussen Maulud et al., "A State of Art for Semantic Analysis of Natural Language Processing," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 21-28, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ritika Singh, and Satwinder Singh, "Text Similarity Measures in News Articles by Vector Space Model Using NLP," *Journal of The Institution of Engineers (India): Series B*, vol. 102, pp. 329-338, 2021. [CrossRef] [Google Scholar] [Publisher Link]