Aggregating Knowledge from Multiple Sources for Estimation of Overlap between Courses

Adwait P. Parsodkar[1,2,*,†], Shanu Kumar[2,†], Muqeeth Mohammed[3,†], Tapish Garg[4,†], Shania Mitra[1,†], Shashank Patil[1,†], Anil Prabhakar[1,†] and Sutanu Chakraborti[1,†]

[1]*Indian Institute of Technology, Madras, Chennai, 600036, India*
[2]*HCL Technologies*
[3]*University of North Carolina at Chapel Hill*
[4]*Honeywell, Bengaluru, India*

Abstract

The courses offered at Universities concerning higher education tend to be highly interdisciplinary, due to which courses across departments are likely to overlap. Therefore, in situations when a new course is to be introduced in a department, it becomes necessary to ensure that the proposed course does not have a significant overlap with any of the pre-existing courses within the same or another department. Since the number of courses offered at universities is large, it becomes difficult to manually inspect all the course contents for overlap. While traditional Information Retrieval systems can help reduce manual efforts, they remain ineffective in incorporating other course-related information, such as that present in the associated list of reference books. In this work, we build an assistance system that attempts to capture the extent of overlap between courses offered at Indian Institute of Technology, Madras (IITM) using world knowledge from multiple knowledge sources such as Wikipedia[1], Google Books[2], etc. This system is intended to add additional courses-related functionality to Workflow, a platform for connecting various departments in the institute.

Keywords

Case-Based Reasoning, Explicit Semantic Analysis, Information Integration

1. Introduction

Institutions of higher education offer a wide spectrum of courses that are often interdisciplinary in nature. That is, a course offered in a department might be closely linked with concepts of another discipline. For instance, *CS6024 - Algorithmic Approaches to Computational Biology* offered at Indian Institute of Technology, Madras[1] (IITM) is closely linked with the two departments - Computer Science and Engineering (CSE) and Biotechnology (BT). As a result of this, while proposing a new course in a department, an exhaustive search over all the courses offered

[1]https://en.wikipedia.org/
[2]https://books.google.com/

ICCBR POCBR'22: Workshop on Process-Oriented Case-based Reasoning at ICCBR-2022, September, 2022, Nancy, France

*Corresponding author.
cs20d404@cse.iitm.ac.in (A. P. Parsodkar); cs19m060@smail.iitm.ac.in (S. Kumar); ee16b026@smail.iitm.ac.in (M. Mohammed); tapishgarg0@gmail.com (T. Garg); shaniamitra9@gmail.com (S. Mitra); shashankpatil0705@gmail.com (S. Patil); anilpr@iitm.ac.in (A. Prabhakar); sutanuc@cse.iitm.ac.in (S. Chakraborti)

[1]https://www.iitm.ac.in/

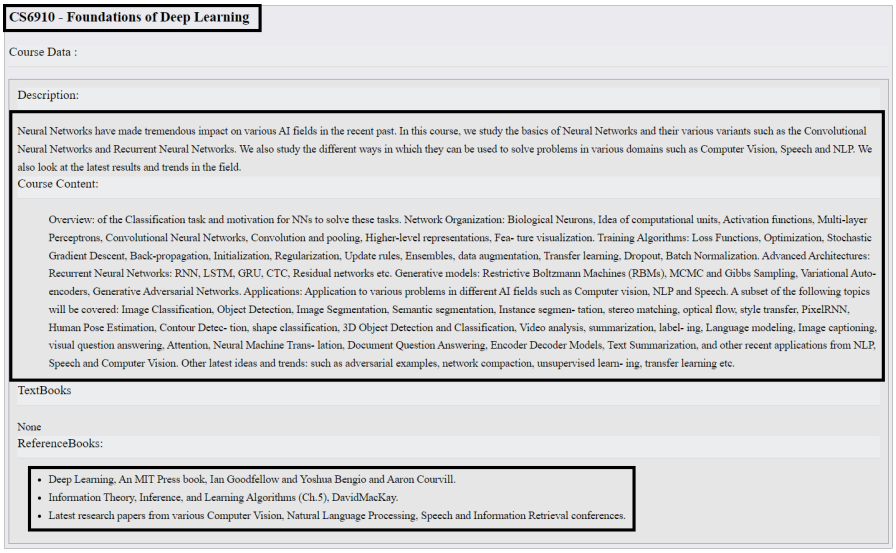


Figure 1: Snapshot of the IITM web page corresponding to *CS6910 - Foundations of Deep Learning*

by the university becomes necessary in order to ensure that the newly proposed course does not have a significant *overlap* with a pre-existing course in the same department or in another. Since the number of courses in the database of the institute can potentially be very large (for instance, 520 courses were offered in IITM in the odd semester of 2019-20 academic year), there arises a need for an automated system that can help retrieve courses that overlap with the input course. Such an assistance system can streamline expert efforts to inspecting course overlap only over a selected top few courses from the ones retrieved by the system.

IITM uses *Workflow*[2], a platform used to connect various departments such as Accounts, Payroll, Academics administration, General administration, etc. The users of such a platform include the scholars, the faculty members (including faculty advisors), and other staff members of the institute. Workflow has been used for numerous academic and administrative tasks, and its functionality can be extended to course-related features too. In particular, we aim at this task of identifying the pre-existing courses in the institute database that have a significant overlap with a new course that a faculty member wants to introduce.

A course is often described using multiple fields, such as its title, the course content, a list of reference books, etc - an example is shown in figure 1[3]. The task of estimating the extent of overlap between courses is challenging because of the following reasons:

- The first is the problem of paraphrasing, where the vocabulary used in describing the content of courses often differs across course descriptions. For instance, the words *feature* and *attribute* are often used interchangeably in the context of courses related to Machine Learning.
- Using straightforward Bag-of-words (BOW) methods for representation of courses might

[2]https://workflow.iitm.ac.in/
[3]https://www.cse.iitm.ac.in/course_details.php?arg=MTUw

result in loss of information conveyed by key *phrases*. This is due to the fact that the combination of meanings of constituent words of a phrase is often not representative of the meaning of the phrase. Consider *Deep Learning,* for instance, where the meaning of the constituents *Deep* and *Learning* taken in isolation does not give rise to the actual meaning of the parent phrase. It is therefore crucial to retain phrases and avoid inferring similarity between courses based on the presence of such constituent words.

- Since the course is defined using multiple fields (title, content, reference books, etc.) of varying nature, it becomes essential to handle them separately. For instance, treating the titles of reference books as plain text might result in the loss of critical information about the contents of the book.

In addition to these challenges, we were required to curate ground truth data that comprises a list of course pairs such that the courses in a pair have overlapping content. For the purpose of experimentation, we created a dataset consisting of courses offered at IITM in the odd semester of the 2019-2020 academic year, with the focus on query courses belonging to CSE and Electrical Engineering (EE) department.

We cast this problem of detection of overlap between courses in a Case-Based Reasoning (CBR) [1][2] framework, where the features corresponding to each course captures knowledge about the course from different sources. The final measure of overlap between two courses is then obtained by taking a weighted combination of all feature-level similarities. The use of a variety of knowledge sources allows us to tackle the aforementioned challenges.

2. Problem Statement

In this work, we want to arrive at a list of courses from the set of pre-existing courses $\mathcal{D}$ offered at IITM that have a significant overlap with a course that is being newly proposed in the institute. We assume that the courses to $\mathcal{D}$ have the following associated information:

1. The course title,
2. Description of the course in terms of the topics covered,
3. A list of reference books, and
4. An optional list of prerequisite courses.

For the sake of experimentation, we adopt the leave-one-out strategy where we consider every existing course belonging to the CSE and EE department as a query (course to be introduced) while treating the remaining set of courses as the corpus to be rank ordered. We further assume that the extent of overlap between courses is quantified by their similarity. The Information Retrieval (IR) system then retrieves courses most similar to the query course and averages the performance over all such queries. We want the ordered list generated by the IR system to place the most relevant courses at the top ranks for every query course.

3. Background

In this section, we briefly discuss the necessary background for techniques used in our approach.

3.1. Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) [3] is an approach that represents text in a space spanned by orthogonal *explicit* concepts. [3] regards Wikipedia articles as the concepts defining the concept space. Any piece of text is therefore placed in the space governed by these Wikipedia articles. Towards obtaining the *concept vector* of words, a standard TF-IDF term-document (which, in this ESA terminology, is a term-concept matrix) matrix is built over a (sub)set of Wikipedia articles $\mathcal{W}$ that are relevant to the domain under consideration. In effect, this matrix expresses every word in the vocabulary of $\mathcal{W}$ in terms of the Wikipedia articles (concepts), and these representations are regarded as the concept vectors corresponding to the terms in $\mathcal{W}$. By construction, the concept vectors of terms that co-occur in the several articles in $\mathcal{W}$ (such as *derivatives* and *calculus* when $\mathcal{W}$ contains Wikipedia articles in the mathematics domain) will be placed close to each other in the concept space. As a result, the domain-specific world knowledge captures relatedness between these terms, which were otherwise considered *orthogonal* in the standard Vector Space Model.

Finally, the concept vector representation of a documents d in a corpus $\mathcal{D}$ is obtained by taking a weighted linear combination of concept vectors of its constituent words weighted by its TF-IDF weight in $d \in \mathcal{D}$. It is to be noted that words in $\mathcal{D}$ that do not appear in the vocabulary of $\mathcal{W}$ do not contribute to the document concept vector computation.

Expressing text as concept vectors using such an approach has the multiple advantages. Firstly, the use of Wikipedia articles enriches the representation with world knowledge present on Wikipedia. In particular, ESA has been shown to capture word relatedness based on the co-occurrence of related words in the same Wikipedia articles. Secondly, these concept vectors are interpretable, in contrast to concept vectors in Latent Semantic Analysis (LSA) [4], where it is difficult to read into the concepts.

3.2. Phrase Extraction

As highlighted in Section 1, the interpretation of a phrase might be significantly different from the composite interpretation of its constituent words taken in isolation. As a result of this, treating phrases as a single entity rather than breaking it into constituent words (in practice, translating the phrase such as *Deep Learning* to the token *Deep_Learning*) might improve effectiveness of retrieval. A list of phrases can be curated in a bottom-up fashion based on the Positive Point wise Mutual Information ($PPMI$) between its constituent words, as proposed in [5]. That is, a collocating word pair (w_1, w_2) is said to constitute a phrase if $PPMI(w_1, w_2)$ is more than a pre-defined threshold θ.

4. Approach

We now discuss our approach to incorporate knowledge from the following knowledge sources while estimating the overlap between courses.

- Domain-specific world knowledge from Wikipedia,
- Domain-specific knowledge from course contents of related courses obtained as available in the web pages of multiple eminent institutions,

- High quality phrases from online glossaries,
- Knowledge about the contents of books from Google Books, and,
- Prerequisite course information.

4.1. ESA over domain-specific concepts

In this section, we first discuss the construction of two domain-specific corpora $\mathcal{W}_{wiki}$ and $\mathcal{W}_{oucc}$ that are separately used to construct two sets of concept vectors using ESA (as discussed in section 3.1). We construct a subset of Wikipedia $\mathcal{W}_{wiki}$ by querying Wikipedia with the titles of courses offered at IITM. From the retrieved results, top$-k$ Wikipedia articles are used to populate $\mathcal{W}_{wiki}$. The creation of such subset of Wikipedia articles serves two major advantages:

- *Improved Efficiency*: The number of dimensions is significantly lower than that when the entire Wikipedia is used, and so the cost involved in similarity computation and storage is lesser.
- *Domain-specific knowledge*: $\mathcal{W}_{wiki}$ by construction comprises Wikipedia articles that are relevant to the domain, and so the background knowledge is highly specific to the domain under consideration.

In addition, a set of concepts W_{oucc} is curated by querying Google[4] and fetching the contents of courses offered in the following institutes: Massachusetts Institute of Technology (MIT), Stanford University, New York University, Indian Institute of Technology(IIT) Bombay, IT Kanpur, IIT Roorkee, IIT Delhi, IIT Kharagpur, Indian Institute of Science Bangalore. Such a set of concepts results in incorporation of knowledge from the course contents from these institutions.

The implementation of ESA was enriched with the knowledge of phrases obtained in both top-down and bottom-up fashion. The top-down phrases were collected from glossary pages on Wikipedia[5], whereas the bottom-up phrases were obtained by considering the top-70 word pairs that had the highest $PPMI$.

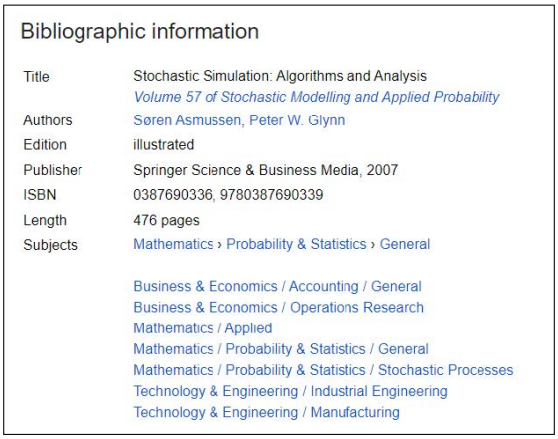
4.2. Computation of similarity between reference books

Since the titles of books are not necessarily good indicators of the content of the books and since we do not have access to the contents of the books, we resort to using the details associated with the books from the Google Books API. Corresponding to each book, a set of words is created that indicate the subject pertaining to the book, the main topics associated with it, and the most frequently occurring words. A sample example is shown in figure 2. Finally, the similarity between the two books is defined as the Jaccard Similarity between the sets corresponding to the two books. That is, given two sets $\mathcal{B}_1, \mathcal{B}_2$ of key terms corresponding to two books b_1, b_2 respectively, the similarity between the two books is defined as,

$$sim_{book}(b_1, b_2) = \frac{|\mathcal{B}_1 \cap \mathcal{B}_2|}{|\mathcal{B}_1 \cup \mathcal{B}_2|}$$

[4]https://www.google.com/

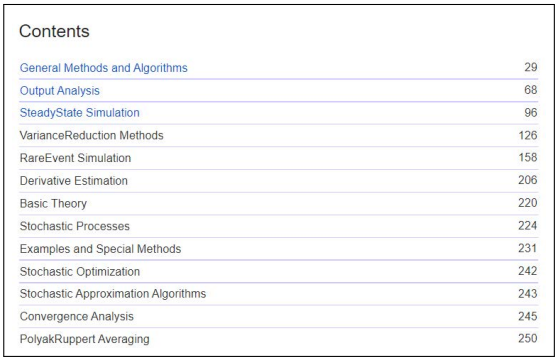
[5]https://en.wikipedia.org/wiki/Glossary_of_computer_science,
https://en.wikipedia.org/wiki/Glossary_of_electrical_and_electronics_engineering



Bibliographic information

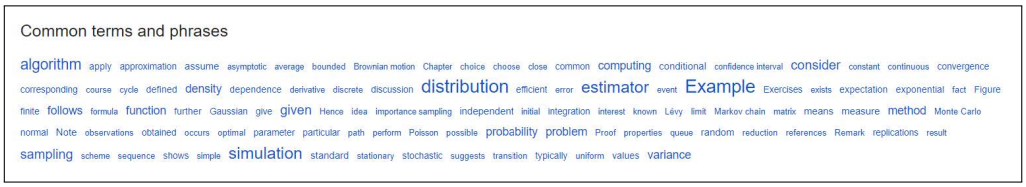
Title	Stochastic Simulation: Algorithms and Analysis *Volume 57 of Stochastic Modelling and Applied Probability*
Authors	Søren Asmussen, Peter W. Glynn
Edition	illustrated
Publisher	Springer Science & Business Media, 2007
ISBN	0387690336, 9780387690339
Length	476 pages
Subjects	Mathematics › Probability & Statistics › General Business & Economics / Accounting / General Business & Economics / Operations Research Mathematics / Applied Mathematics / Probability & Statistics / General Mathematics / Probability & Statistics / Stochastic Processes Technology & Engineering / Industrial Engineering Technology & Engineering / Manufacturing

(a) General Information about the book



Contents

(b) Contents of the book



Common terms and phrases

algorithm apply approximation assume asymptotic average bounded Brownian motion Chapter choice choose close common computing conditional confidence interval consider constant continuous convergence corresponding course cycle defined density dependence derivative discrete discussion distribution efficient error estimator event Example Exercises exists expectation exponential fact Figure finite follows formula function further Gaussian give given Hence idea importance sampling independent initial integration interest known Lévy limit Markov chain matrix means measure method Monte Carlo normal Note observations obtained occurs optimal parameter particular path perform Poisson possible probability problem Proof properties queue random reduction references Remark replications result sampling scheme sequence shows simple simulation standard stationary stochastic suggests transition typically uniform values variance

(c) Common terms and phrases in the book

Figure 2: Information associated with *Stochastic Simulation* book from Google Books.

4.3. Computation of similarity between courses based on Prerequisite information

Given two courses $c_1, c_2 \in \mathcal{D}$, and corresponding sets of prerequisite courses $\mathcal{P}_1, \mathcal{P}_2$ respectively, the prerequisite similarity between c_1 and c_2 is defined as,

$$sim_{prereq}(c_1, c_2) = \begin{cases} 1 & \text{if } c_1 \in \mathcal{P}_2 \text{ or } c_2 \in \mathcal{P}_1 \\ 0 & \text{otherwise} \end{cases}$$

Note that the case where the two courses have a common prerequisite, but one is not a prerequisite of the other results in sim_{prereq} score of 0 since sharing a common prerequisite course may not necessarily indicate an overlap between the courses. For instance, a basic *Probability Theory* course might be a prerequisite for a *Machine Learning* course and a *Randomized Algorithms* course, however the two courses themselves do not overlap in content.

4.4. Course overlap estimation in CBR

To integrate knowledge from the multitude of knowledge sources into the course overlap estimation process, we treat each course as a case in the case base. The features that describe a case (course) correspond to the representation of the course according to the various knowledge sources. In order to compute the local (feature-level) similarity, we use the *cosine* similarity

function for the ESA concept vectors constructed over $\mathcal{W}_{wiki}$ and $\mathcal{W}_{oucc}$, while Jaccard similarity is used to compute similarity between set representation of books. A weighted combination of these local similarity values gives the course-level similarity for the two input courses.

5. Dataset and Experimental Results

5.1. Dataset

In this work, the set of courses $\mathcal{D}$ is constructed by considering the offerings at IITM in the odd semester of 2019-2020 academic year. $\mathcal{D}$ consists of 520 courses across all the departments at IITM. In this work, we restrict our attention to cases where a query course q is one of the 87 courses - 33 belonging to the CSE department or 54 from the EE department. The ground truth was constructed manually and is represented as a binary-valued 87×520 matrix $\mathcal{G}$ such that

$$\mathcal{G}_{i,j} = \begin{cases} 1 & \mathcal{D}_i^{CSE \cup EE} \text{ overlaps with } \mathcal{D}_j \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{D}_i^{CSE \cup EE}$ represents the i^{th} course in the subset of $\mathcal{D}$ comprising courses belonging to the CSE and EE department, and $\mathcal{D}_j$ represents the j^{th} course in $\mathcal{D}$.

For the construction of the domain-specific subset of Wikipedia, $\mathcal{W}_{wiki}$, all the 520 course titles were queried to Wikipedia, and the top-15 retrieved articles corresponding to each query were used to populate $\mathcal{W}_{wiki}$. This resulted in 5287 unique Wikipedia articles in $\mathcal{W}_{wiki}$.

5.2. Evaluation and Results

To quantify the effectiveness of our system, we use the normalized DCG measure [6] where the DCG score associated with a ranking of courses similar to the j^{th} course, r_j, is given by,

$$DCG(r_j) = \sum_{i=1}^{|r_j|} \frac{\mathcal{G}_{j,r_j[i]}}{log_2(i+1)}$$

where $r_j[i]$ is the index of the i^{th} course in the ranking r_j.

The extent of closeness of the DCG score with the DCG of the ideal ranking (quantified by normalizing with the DCG of the ideal ranking) gives the $nDCG$ score for a ranking.

The overall performance of the system is given by averaging the $nDCG$ score of the rankings over the set of courses $\mathcal{D} - q$ for the every $q \in \mathcal{D}^{CSE \cup EE}$.

The weights associated with the 4 knowledge sources were obtained by maximizing the average $nDCG$ score using cross-validation over a random subset comprising 80% of the CSE courses as training data using a grid search. The effectiveness of the different models is summarized in table 1. Due to the sparseness of prerequisite and book information associated with the courses, these knowledge sources tend to perform poorly in isolation. However, when the knowledge from prerequisites and books is combined with ESA that uses $\mathcal{W}_{wiki}$ and $\mathcal{W}_{oucc}$, the average $nDCG$ is found to improve.

Table 1
Average nDCG for different models.

Sr. No.	Model	average nDCG
1.	ESA with $\mathcal{W}_{wiki}$	0.771
2.	ESA with $\mathcal{W}_{oucc}$	0.781
3.	Combined	**0.811**

5.3. The User Interface

As a part of this work, we also constructed a User Interface (UI) as shown in Figure 3 that takes as input a query course and the local similarity weights associated with the four knowledge sources and produces a list of overlapping courses. The UI is intended to experiment with different weights corresponding to the knowledge sources and to analyze its effect on the retrieved results. The UI also shows a redundancy measure that attempts to answer the following question: With respect to a certain course that a student may have taken, what percentage of a new course being proposed becomes redundant? However, in the context of the results shown, this option has been disabled. Further, he UI highlights the important overlapping words between the query course and retrieved courses. For example, Figure 4 shows the words in common between the query 'CS6852 - Theory and Applications of Ontologies' and the retrieved course 'CS5102 - Topics in Semantic Web Technology'.

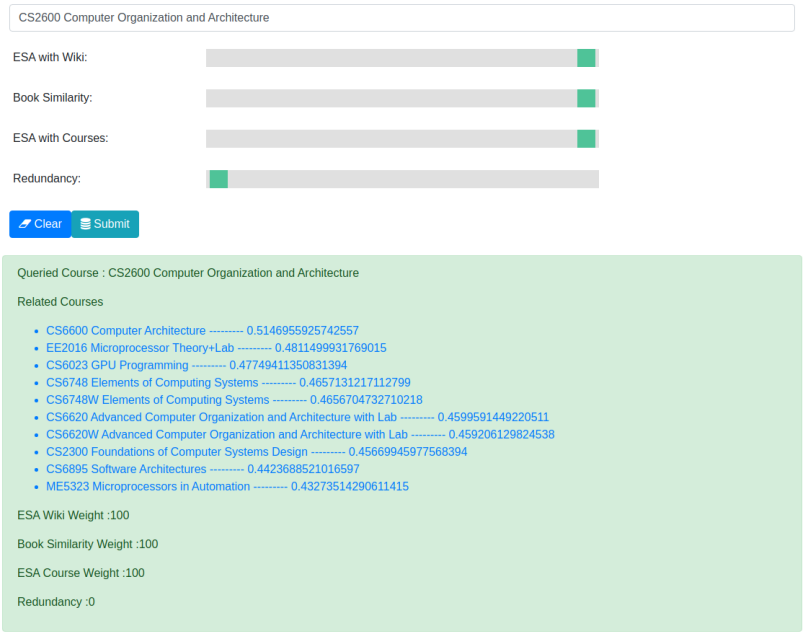


Figure 3: Demonstration of the UI with *CS2600 Computer Organization and Architecture* as the query course, and the local similarity weights for the respective knowledge sources being $(1/3, 1/3, 1/3, 0)$.

CS5102 Topics in Semantic web Technology

Topics in Semantic web Technology Several re search issues concerning enhancement of DLs to handle new requirements such as temporal aspects, process Description s, Ontology based information systems and information integration are currently underway in the Semantic web community. Selected items from these would be taken up in depth. A list of topics is given below: Kn OWL edge graphs - their construction, Query ing and deployment when there is there is huge amount of Data available. SPARQL Query processing: SPARQL is a Query language for RDF Data and building RDF Data stores and efficiently running SPARQL queries is an ongoing re search direction. Reasoning systems : Reasoners for DLs are systems that infer or check the validity of Inference of a given fact with respect to given Ontology . Re search ing on new techniques for efficient Implementation of reasoners and devising optimisation techniques would be a topic of interest. OBDA(Ontology Based Data Access): Issues in realizing Ontology based information systems and their implementation. Omtologies and NLP: Simpler, more natural language based interfaces for authoring ontologies is Need ed for wide-spread use of Ontology framework s. Natural language Description s of formal ontologies (or parts of them) are useful for non-cs experts to understand the Kn OWL edge represented. Automated text generation is high interest here. The proposed course offers scope for in-depth study of the recent topics - such as those mentioned above - in the Semantic web Technology and would be largely seminar based. After an initial set of lectures that set the context, selected papers from International Semantic web Conference (ISWC), Extended Semantic web Conference (ESWC), Semantic web Journal (SWJ), Journal of Web Semantic s would be studied by student teams and presentations would be made.

Figure 4: Example of important words overlapping between the query 'CS6852 - Theory and Applications of Ontologies' and the retrieved course 'CS5102 - Topics in Semantic Web Technology'.

6. Future Work

ESA suffers from a fundamental limitation of assuming the concepts being orthogonal to each other. However, they are not necessarily orthogonal - Machine Learning[6] and Deep Learning[7] articles, for instance, are highly related. Non-Orthogonal ESA (NESA) [7] has made attempts to capture relatedness between these concepts. However, it has not been explored in our work.

Feedback from the users of the system is also an important source of knowledge. This can feed into the adaptation phase so that alterations can be made to the generated ranking in order to improve the retrieval effectiveness with increasing number of feedback. We have performed preliminary work towards incorporating feedback from users using Retrofitting [8] over the courses in the CSE department. The underlying idea of Retrofitting is to modify the vector representation of two (positively) related entities (as learnt from positive feedback provided by the user) such that the new representations are placed closer to each other without moving them much farther from their original positions. While we have observed improvement with incorporation of feedback from users over the courses offered in the CSE department (Figure 5), exhaustive experiments are to be conducted to provide robust results.

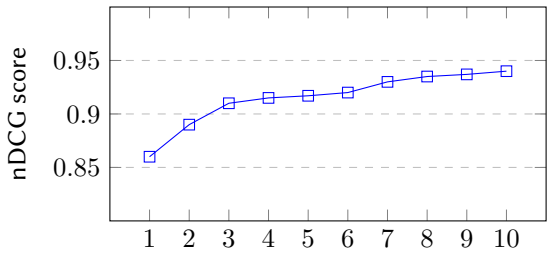


Figure 5: Improvement in the nDCG score with the number of modifications of ESA vectors over $\mathcal{W}_{wiki}$ using Retrofitting

Finally, the current work is only concerned with a particular use case relating to courses in a university. There exist other use cases that associate with more stakeholders that remain

[6]https://en.wikipedia.org/wiki/Machine_learning
[7]https://en.wikipedia.org/wiki/Deep_learning

unexplored. For instance, an assistance system that flags a warning to a faculty advisor when a student attempts to take a course that is redundant given the student's course history, thereby preventing free credits to the student. Another example could be a course recommendation system that suggests courses that a student can credit given the courses the student has completed in the past, and the specialization she wants to end up with at the end of her course-work.

7. Conclusion

In this work, we have demonstrated how the CBR framework can allow integration of knowledge about the same entity from different knowledge sources. In particular, we focused on using domain-specific world-knowledge from Wikipedia, highly relevant knowledge from course contents from various universities, prerequisite course information, and book information from Google Books for the estimation of overlap between courses. The combination of these knowledge sources were found to be more effective than when each of these knowledge sources are used in isolation.

Acknowledgments

We would like to thank Devi Ganeshan, Neha Kuntewar, Meera Serawat, Muhammed Shamil, Praveen Kumar, Anand, and Naveen Kumar for assisting with the creation of the ground truth, and to Manohar Venugopal, Sachin Virmani, Surya Kiran, Ramesh Soni and Monika Gupta, for actively working with us on this project.

References

[1] J. L. Kolodner, An introduction to case-based reasoning, Artificial intelligence review 6 (1992) 3–34.

[2] A. Aamodt, E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches, AI communications 7 (1994) 39–59.

[3] E. Gabrilovich, S. Markovitch, et al., Computing semantic relatedness using wikipedia-based explicit semantic analysis., in: IJcAI, volume 7, 2007, pp. 1606–1611.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American society for information science 41 (1990) 391–407.

[5] K. Church, P. Hanks, Word association norms, mutual information, and lexicography, Computational linguistics 16 (1990) 22–29.

[6] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems (TOIS) 20 (2002) 422–446.

[7] N. Aggarwal, K. Asooja, G. Bordea, P. Buitelaar, Non-orthogonal explicit semantic analysis, in: Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, 2015, pp. 92–100.

[8] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, N. A. Smith, Retrofitting word vectors to semantic lexicons, arXiv preprint arXiv:1411.4166 (2014).