Learning Object Recommendation through Knowledge Graph

Taranveer Singh **Saksham Singhal** **Tejas Nama** **Tiancheng Liu** **Yuqi Zhang** **John Stamper** (Advisor)

1 Introduction

1.1 Concept Graph

As we are making exponential progress in different areas of science, it is hard for students, researchers, teachers and educational institutions to catch up. With the rise of learning content shared across online MOOCs, Wikipedia, StackOverflow, Scientific Journals and Quora, we have abundance of information which presents the same content in varied depths and points of view for different types of learners. To enable these learners to fulfill their education goals with ease, there is a need to leverage the vast range of learning materials on the web. The first step towards solving this problem is to define the granularity and hierarchy of concepts, and then organizing concepts in a graph based on the sequence of prerequisites and post-requisites which will help structure the learning content to cater to different types of learners. Most of the recommendation systems in various domains recommend content based on similar items a user has seen in the past or is currently browsing, but education domain is different. A knowledge graph of concepts has various potential use-cases like automatic answer generation to resolve specific learner doubts by leveraging learning resources, educational search engine, and a simpler application would be forum recommendation.

1.2 Forum Recommendation as an Application

Forum engagement has been proven to improve learning of students in MOOCs and also, people have found a high correlation with the participation in forum discussions and forum activities and the completion of course [1]. Instructors also know a lot about the misconceptions about students from forum, and use it to improve the course materials [2]. However, with 40k+ enrolled students in a MOOC, forums tend to clutter over a period of time which leads to the decline in user engagement. Hence, there is a need for a tool that will leverage forums to help both learners and instructors. Prior research highlights that people look into forum when they are struggling with a certain concept in the assignments or videos, and they go back and forth between forums to resolve their doubts [3]. So there is also a need for a tool that will help understand user intent for better forum recommendations. Forum Recommendations tried before is based on matrix factorization approach [4] and topic modeling based approach [2] to model user forum interaction as we did in Spring semester. They need a lot of user forum interaction data to get relevant results which is not possible in a MOOC. 1) There is a cold start problem and only work for students who have high activity on forums. 2) MOOCs discuss different content on a week by week basis. Hence recent activity is more relevant than past activity which is not taken into account into system proposed by Yang et al [5]. 3) These approaches are based on topic modelling which operate on a latent space and hence, don't help the instructors gain insights via forums. Forum Recommendation needs to improve over time.

2 Goals / Hypothesis

In order to build a recommender system for learners we have the following set of goals.

1. Understand learner's contextual queries/posts and be able to map it to concepts.
2. Build a concept graph capable of representing syntactic similarity as well as pre-requisite relationship between concepts.

To tackle these problems we intend to use information from different sources capturing different levels of concept complexity. Also, we will evaluate different components of concept graph on the task of forum recommendation for MOOC's.

3 Methodology

The flowchart in the Figure 1 shows are our entire pipeline in a step by step manner as described in the goals above. Starting from incorporating different sources, performing concept extraction followed by concept inference using word and document embeddings and finally concept-graph based recommendation. Concept-Graph is one of the important highlights and we define it as a directed acyclic graph which represents the distance between concepts and hierarchy of concepts in a particular domain. Each node in the graph is a canonical, discriminant concept in the domain of interest and the links between nodes indicate the pre-requisite and post-requisite relationships. We aim to use external knowledge from textbooks, video transcripts and Wikipedia to generate concept graph.

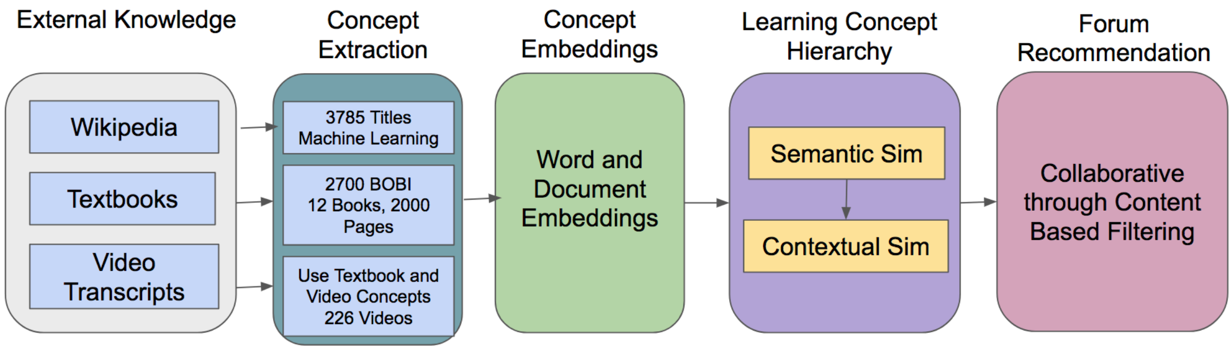


Figure 1: Flowchart of Proposed Mehtodology

Concept Extraction The first step in constructing the concept graph is to extract concepts from the data (concept extraction). The granularity level of these concepts are very important to get a good representation. We hypothesize that Wikipedia and textbooks can provide a rich set of words that covers the entire domain. We also attempt to extract these concepts by parsing each sentence and the head word of the noun phrase. We observe that the former approach works better and are able to extract meaningful concepts.

Concept Inference The second step is to infer latent concepts from a forum post or a user query i.e concept he is interested in misconception he has. This method includes dimensionally reduced word vectors through which we can represent both concepts and the articles in the same space and then find the closest concepts as illustrated in the Figure 2. We train the word2Vec model on the external data corpus consisting of wikipedia articles, textbooks and video transcripts to get the concept embeddings. Additionally, we also train a SIF embeddings [6], a document embedding model, on the same corpus with word2vec model.

Feature Extraction The next step in building the concept graph is to derive features which represent the pre-requisite relationship between the extracted concepts. We use three types of features:

1. Semantic features - Describes the semantic distance between concepts. It represents the probability of two concepts being used in the same context.
2. Contextual features - Describes the prerequisite relationship between concepts based on context information.
3. Structural Features - Describes the structure of the document, and considers the order in which concepts occur.

Using these features we train a linear model to predict the relationship (pre-requisite or post-requisite) between two concepts and generate the concept graph.

Forum Recommendation The last part of the system is to make relevant predictions to a particular user based on user-forum activities. We also use the features obtained from the constructed concept graph trained on the required domain. Most of the recommender systems recommend articles or forums based only on user similarity or item similarity or both [7]. But with the extra priors extracted from the concept graph, we can also make predictions which are in the prerequisite and post-requisite space. Moreover, Concept-Hierarchy model uses explicit features and this helps us to qualitatively evaluate the rationality behind predictions.

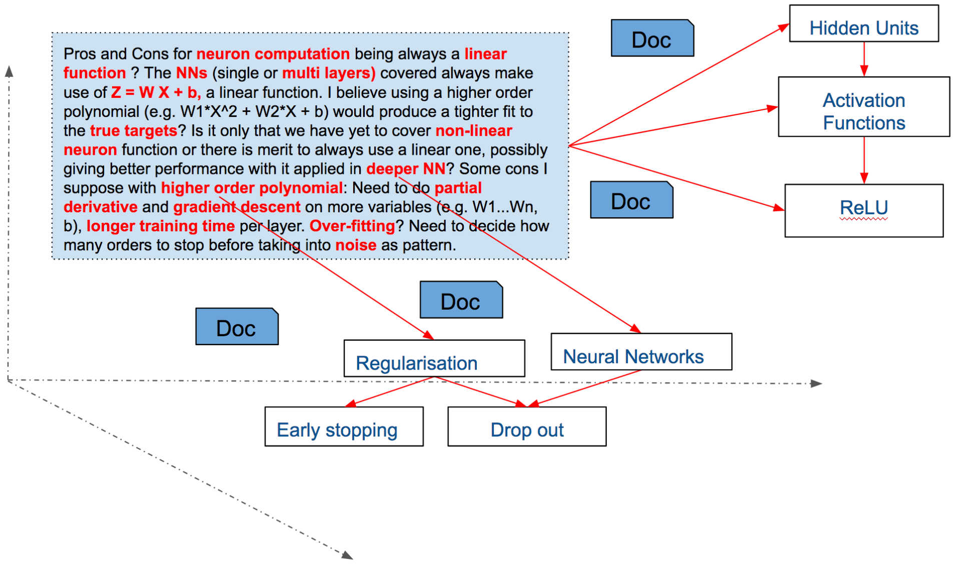


Figure 2: Concept Inference

4 Experimental Design

4.1 Word2Vec with Different Sources

About the Dataset: We train the word2vec model with three different external sources of data to capture different levels of concept complexity.

- Wikipedia - We scrape data from wikipedia and collect only the articles which are related to the machine learning. The extracted wikipedia corpus consists of about 3785 articles.
- Textbook - We collect 12 books in machine learning domain, consisting of about 2000 pages.
- Videos - We also collect video transcripts from 1914 videos of machine learning online courses.

Model: Since sources capture different concept complexities, training a single word2vec model by combining data from all the sources would possibly result in losing relevant information. Training three different models for each source would help avoid this problem, but we would require all the concepts to be in the same space for extracting prerequisite features. Thus we train one model by combining datasets from different sources and tagging the concepts with a source indicator. For example, a concept like "gradient descent" in wikipedia gets tagged as "w_gradient_descent_w" indicating the source. This ensures that a single source does not dominate over other sources and we retain the important information. We have illustrated this idea in Figure 4 where we compare the average complexity of concepts in different sources. The complexity of a concept is defined by it's idf score. We finally conclude the complexity of a concept is greater in wikipedia than textbook followed by video.

Results and Error Analysis: We conduct qualitative analysis and human test evaluation to check the performance of Word2Vec models trained separately over three different sources.

- Human Evaluation: The task consists of 100 questions, where each question had a base word and four options. We (test was taken by the five team members) had to choose an option which was the farthest from the base word. We see that all the three models performed better than random (= 25%). In order to evaluate it we plotted the concepts using TSNE and performed analogy tests as shown in figure 3.
- Complexity of Concepts: It is reasonable to assume that complexity of a word is inversely proportional to the number of times it occurs in the corpus. Hence we compute the average idf of the top 50 words closest to a given concept as the complexity of the concept. Table 4 shows examples of "gradient descent" and "svm", and we clearly see that the idf and hence complexity of wikipedia is greater. The figure also shows the closest concepts to "gradient_descent" for a particular source which helps in qualitatively evaluating the complexity of source.

Figure 3: Qualitative Analysis of Word2Vec

WIKIPEDIA	TEXTBOOK	VIDEO
w_gradient_descent_w: 5.66	**t_gradient_descent_t: 4.92**	**v_gradient_descent_v: 4.86**
w_gradient_w	t_stochastic_gradient_descent_t	v_cost_function_v
w_stochastic_gradient_descent_w	t_backpropagation_t	v_algorithms_v
w_iteration_w	t_perceptrons_t	v_convergence_v
w_backpropagation_w	t_update_t	v_update_v
w_hill_climbing_w	t_gradient_t	v_stochastic_gradient_descent_v
w_error_function_w	t_line_search_t	v_learning_rate_v
w_contrastive_divergence_w	t_error_function_t	v_gradient_v
w_steepest_descent_w	t_delta_rule_t	v_derivative_v
w_softmax_function_w	t_steepest_descent_t'	v_implementations_v

Concept	Wikipedia	Textbook	Video
gradient_descent	5.66	4.92	4.86
SVM	6.49	5.38	5.24
All Average	4.2	1.6	1.1

Complexity: Wikipedia > Textbook > Video

Figure 4: Analysis of complexity of Different sources

4.2 SIF Embedding

About the Dataset: The data required for this model is the concept vectors which are trained using word2vec model as described in the above subsection. We also need the raw text and the granularity of division of the text to be used as sentences in our model.

Model: SIF stands for the 'smoothed inverse frequency', which is an unsupervised scheme for generating sentence embeddings [6]. It has been shown to consistently outperform simple BoW approach and also sentence vectors trained using LSTM models at many intrinsic and extrinsic tasks. SIF model is slightly more complicated than the simple averaging of BoW vectors where we multiply each component vector by the inverse of its probability of occurrence as follows

$$v_s \rightarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a + p(w)} v_w$$

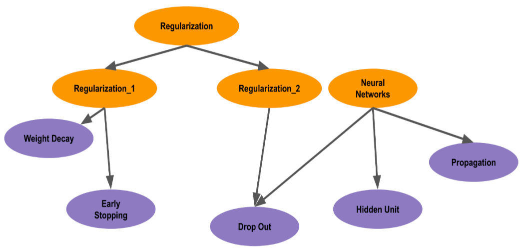
This is followed by performing SVD over the computed sentence embedding matrix and subtracting the projection of the sentence embedding on the first computed principal component of the matrix. This results in the removal of most frequent and syntactic discourse bias from the sentence vectors.

4.3 Contextual SIF Embeddings

Model: Even though the source tagged word2vec model described in section 4.1 is able to retain context of concepts which belong to the same source, it is still possible that the given concept is used in different contexts and the model fails to capture such scenarios. Word sense disambiguation is an open research problem in NLP to identify the sense in which such polysemous words are used. It has been shown that the embedding derived from the word2vec model is the linear combination of all the senses of the word [8]. The authors use sparse coding to retrieve different senses of the word.

$$v_{regularization} = \alpha_1 v_{regularization\_1} + \alpha_2 v_{regularization\_2} + \ldots$$

The above equation shows that the vector embedding of the concept ”regularization” can be split into multiple embeddings each describing a different context (discourse). Polysemy will help us disambiguate different contexts of a concept, for example figure 5 lists the closest words to ”regularization” extracted from each of its discourse.



Different discourses for regularization

loss_function	dropout	linear_model
norm	pre_training	tuning_parameter
hinge_loss	network_architecture	linear_regression
squared_error_loss	momentum	ridge_regression
loss	feedforward_neural_network	logistic_regression
penalization	denoising_auto_encoder	lasso

Figure 5: Polysemy for concept ”regularization” different context

Contextual SIF is an extension to the SIF Embedding model which uses ”word-disambiguation” to choose the right word embedding based on the context in which it is talked about. The figure 6 below shows an example where ”run” is defined in three different contexts “running” a computer program, marathon or a president. Through sparse decoding we get three different contexts of the word ”run”. Since SIF is an iterative process, we compute the sentence embedding until each word, “w”, which gives the current context. Based on the distance of the current context to the different sense embeddings of word ”w”, we pick the closest one assuming the the sentence is talking about this context of the word. Since context of a word in a sentence depends on both before and after the word, we compute the entire sentence embedding in both ways and then average the embeddings of sentence.

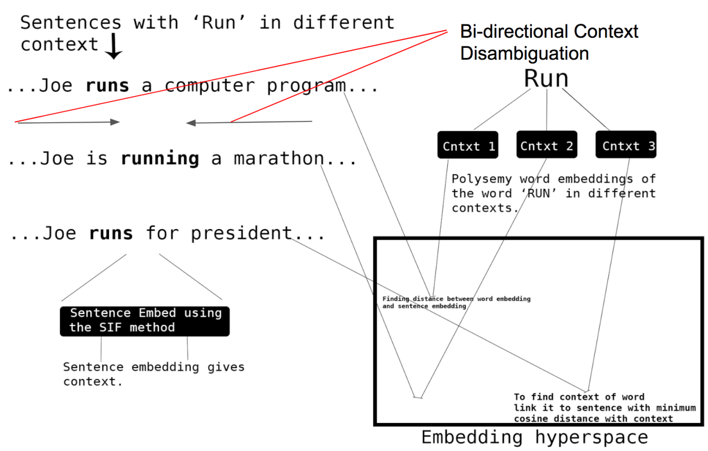


Figure 6: Contextual SIF Model - Concept Disambiguation

Results and Error Analysis: Both SIF and Contextual SIF provide significant improvement over the generic Doc2Vec model, both qualitatively and quantitatively. We perform a simple experiment to test our model (Contextual SIF) by picking random pieces of text pertaining to a certain concept and then extracting the sentence embedding of that text. We then search the rank position at which the concept appeared.

We see that, on average, the concept rank for SIF and Contextual-SIF models are always better than Doc2Vec model [7] and in most cases our model performs better. It is still unsettling that the top concept is not in the top position always. But that does not necessarily have to be true as we can observe in the following qualitative results [8] where even though the exact concept in not in first position but all top ones are highly relevant. (Red denotes irrelevant concept)

Document

Texbook: **We begin by considering the problem of identifying groups, or clusters, of data points in a multidimensional space. Suppose we have a data set $\{x_1, \ldots, x_N\}$ consisting of N observations of a random D-dimensional Euclidean..**

Wikipedia: ***k*-means clustering** is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition ……

Videos: In the clustering problem we are given an unlabeled data set and we would like to have an algorithm automatically group the data into coherent subsets or into coherent clusters for us. The K Means algorithm is by far the most popular….

Rank of concept of document being inferred									
Concepts	Books			Wikipedia Content			Forum/Video		
	Our	SIF	D2V	Our	SIF	D2V	Our	SIF	D2V
Support Vector Machines	34	80	40	34	27	261	17	65	63
Gradient Descent	1	6	25	1	2	283	3	1	80
Dimensionality Reduction	1	2	86	3	1	755	34	32	159
K-Means Clustering	15	13	1	2	17	206	43	4	2
Neural Networks	1	2	5	8	16	40	6	20	390

Figure 7: Rank of Concept Inferred from a Learning Object/Document

Text	Model	Books	Wikipedia	Video Trans.
Gradient Descent	Doc2Vec	committee conjugate_gradients weight_vector overfitting loss	steepest_descent global_minimum geodesic_distance setseed beam_search	setseed local_likelihood trace_operator expectation_propagation for_loop
	SIF Model	gradient update step weight_vector gradient_descent	gradient gradient_descent iteration local_minimum steepest_descent	gradient_descent update weights epoch insurance
	Our Model	gradient_descent update local_minima step conjugate_gradients	gradient_descent newton_raphson coordinate_descent steepest_descent gradient	time dot_product gradient_descent constants propagation

Figure 8: Concept Inferred from a Learning Object/Document

4.4 Concept Hierarchy

About the Dataset: The concept hierarchy is built on 17 coursera courses on the 'Machine Learning' domain with 1914 videos in total. We also use the book corpus and wiki corpus on the same domain described before.

Model: Based on the representation learning-based method proposed in [9], the potential prerequisite relations between knowledge concepts can be learned with contextual and structural information in MOOC courses. Contextual Features help in understanding the prerequisite relations between concepts. Given concept a and b, if when concept a is taught, concept b is also mentioned, but not vice versa. Then, it is very possible that concept b is a prerequisite of a. It could be also extended to book corpus and wiki corpus. Based on the idea, the following indicators are used to infer prerequisite relations between concepts.

1. (Generalized) Video Reference Distance: is the difference of video reference weights which are to quantify how a concept is mentioned in videos about the other concept. In order to deal with the sparsity, the generalized reference weight is proposed by taking similar concepts of the given concept and their similarities with the other concept into consideration.
2. (Generalized) Sentence Reference Distance: is similar as the feature above, but on a sentence level instead of the video level.

The structure of a course (Structural Features) can also contribute to learning the relations of course concepts based on the assumption that concepts are introduced in a course based on their learning dependencies and complexities.

1. Distributional Asymmetry Distance: is inferring prerequisite relations with the course learning dependencies. If a concept a is prerequisite of another concept b, then it is more frequent that concept a is mentioned in its subsequent videos than concept b is mentioned in its precursor videos.
2. Complexity Level Distance: assumes the prerequisite concepts tends to be more basic. Meanwhile, basic concepts tends to appear in more videos and survive longer than advanced ones. We can then get the complexity level of a concept with its average video coverage and average survival time.

Results and Error Analysis: Since we do not have tagged data to check the accuracy of the concept graph, we evaluate the results qualitatively by checking the retrieved prerequisite with Metacademy corpus [10] as show in the figure 9. In general, these features provide sensible results and the structural features perform better than the contextual features do.

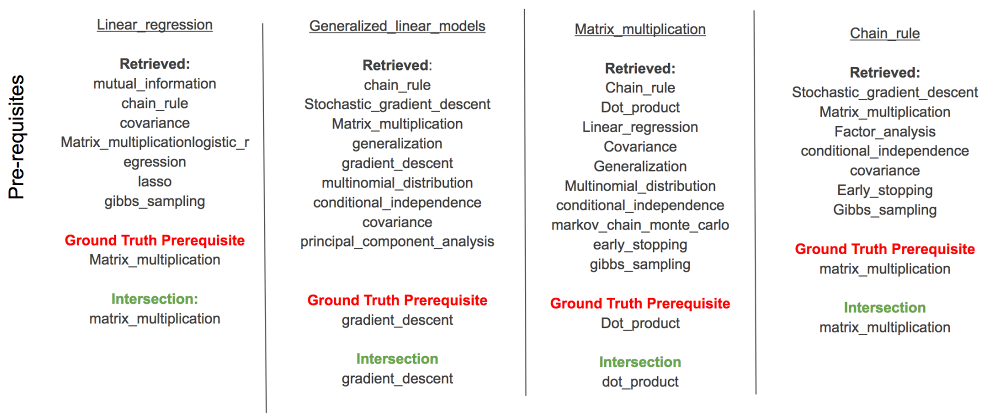


Figure 9: Qualitative Evaluation of Concept Hierarchy compared with Metacademy

4.5 Recommendation

About the Dataset: Since, we haven't yet received forum data from coursera[11], we decided to proceed with a public dataset of CiteUlike [12] scientific articles which holds similar assumptions to forum posts in MOOCs in terms of using Concept Hierarchy for recommendation. The data includes users and their libraries of articles obtained from CiteULike At CiteUlike. Registered users create personal reference libraries; each article usually has a title and an abstract. We merge duplicated articles, remove empty articles, and remove users with fewer than 10 articles to obtain a data set of 5,551 users and 16,980 articles with 204,986 observed user-item pairs. (This matrix has a sparsity of 99.8%; it is highly sparse.) On average, each user has 37 articles in the library, ranging from 10 to 403. 93% of the users have fewer than 100 articles. For each article, we concatenate its title and abstract. We remove stop words and use tf-idf to choose the top 8,000 distinct words as the vocabulary. This yielded a corpus of 1.6M words.

Train Test Division: We randomly take 20% of the articles that users have interacted with, make them as our dev data and exclude them from the training data. In the process of random selection, we make sure that none of the posts get completely excluded from the training data and also it maintains a minimum count of k users so that our recommendation baselines has sufficient information about rating pattern for every posts.

Concept-Graph based Recommendation Model
Figure 10 shows the flowchart for forum/ scientific article recommendation using the concept graph.
Recommendation Resources: From the dataset described before, this simple collaborative filtering recommendation

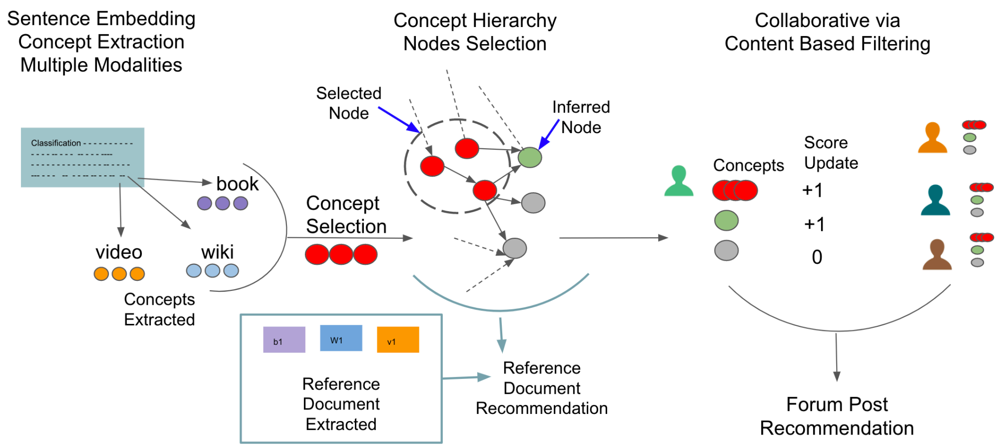


Figure 10: Flowchart of Recommendation Policy using Concept Graph

module seeks viewing behaviors from all users. Also, to use the concept graph, the module uses the list of concepts and the weight mapping from posts to concepts. Weights used in the mapping can be any real number but they must sum to one for every posts, or normalization needs to be applied to enforce this requirement.

Aggregating User Interest: The first step of recommendation is to build a user profile from the browsing behavior. Given the list of posts the target user has viewed, and the mapping from each post to all concepts (Concept Extraction), the module builds a user interest vector by summing the mapping from all viewed posts (Concept Selection). In this case, the user profile vectors are not normalized by number of posts they have viewed because the module does not compare user profile vectors. Since we know the concepts which were viewed by the user, the sparsity of the user profile vector is reduced by including the prerequisite and the postrequisite concepts derived from the concept graph (inferred nodes).

Ranking Unseen Posts: After the user profile vector is built, all unseen posts are ranked according to their similarity scores with the user profile vectors. The scores are calculated using cosine similarity. Then all unseen posts are returned in the descending order on their scores as the final recommendation sequence for this module.

5 Evaluation Metrics

We evaluate our recommendation algorithm on sets of held-out users and related posts to them. We will (hypothetically) present each user with M posts sorted by their predicted rating and evaluate based on which of these posts were actually in the user's engagement list. Engagement list is built by different levels of activities - post, comment, upvote, downvote, view. Two possible metrics are precision and recall. But zero ratings are uncertain. That may indicate that the user did not engage with a post because either no interest or no exposure. This makes it difficult to accurately compute precision. Since all engagements are known to be true positives, we focus on recall. Recall only considers the posts the customer has engaged in within the top M. The objective is to build a system with higher recall and lower M. For each user, the definition of recall@M is

$$recall@M = \frac{\text{number of posts the user is engaged in top M}}{\text{total number of posts the user is engaged in}}$$

The recall for the entire system can be summarized using the average recall from all users. This definition is user oriented. In order to compare with the baseline algorithms like popularity based recommendation, where there is no division of train and dev set, the recall is calculated over all the posts the user is engaged in, including training and dev set.

6 Results and Error Analysis

Please refer to Experimental Design section for results, error analysis, and discussion for each individual module in the system.

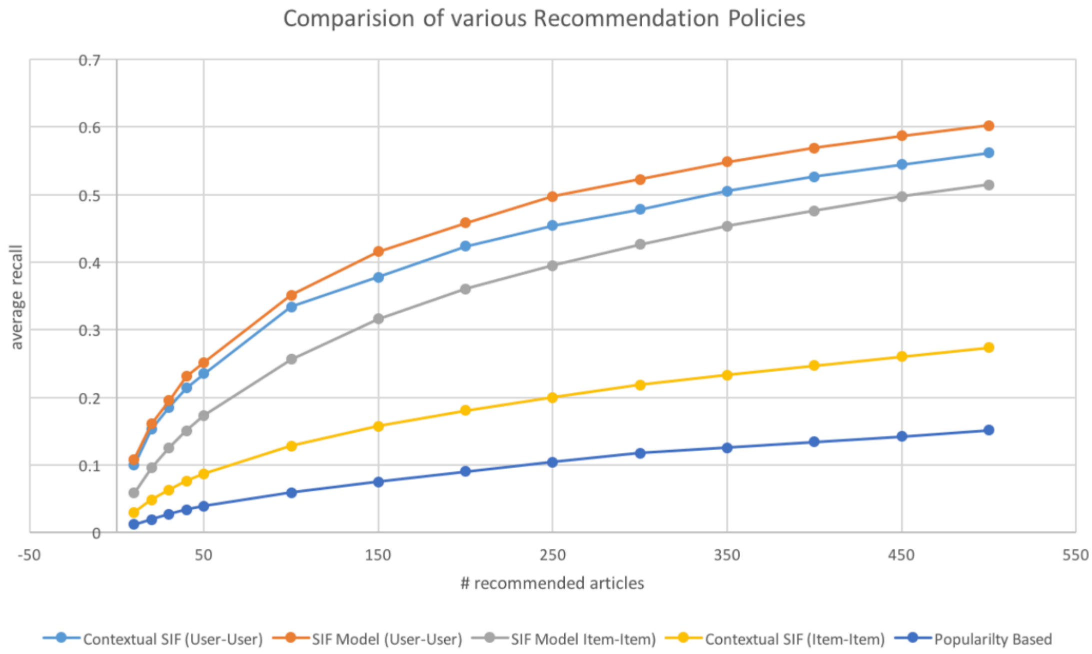


Figure 11: Results of various recommendation policies

From the Figure 11. All proposed methods perform better than popularity based method

1. For User - User based similarity based on concepts inferred from SIF based model. We observe that Contextual SIF performs at par with SIF since scientific articles don't have too many overlapping concepts for polysemy to work.
2. For Item - item based similarity based on just the sentence embeddings from SIF. Contextual SIF performs a lot worse than SIF as, due to varied domains in our dataset of scientific articles polysemy added a lot of noise.

6.1 Discussion

While dividing our dataset for train-test split, we took 20% of the posts for each user as test. On an average, a user has 37 articles on split in test data a user would have about 7 articles, when we recommend by selecting 50 posts out of 16980 posts, we are able to predict 2 out of 7 posts for every user based on a recall of 0.27@50 for our best model. Hence, our algorithm is able to capture signal from the concepts inferred through SIF and Contextual SIF. Though Recall@50 looks low, but our final recommendation comes from 16980 articles and our end goal in mind is to recommend components of learning materials we like to call learning object, so that user doesn't have to choose from the recommended 50 articles but we recommend him the most relevant part from each of the 50 leaning materials selected arrange it in the right sequence based on hierarchy like an automated tutorial. Also, please note that we were not able to test it on the" Applied Machine Learning Coursera forum" dataset, we intended to make our model work towards from the start and hence we did not make use of the Concept Hierarchy which was one of the central part of our Concept Graph. The CiteULike dataset was not what an ideal learning material would look like, we could not use our concept-hierarchy to the fullest extent neither we could use the external knowledge from wikipedia, textbooks and videos as CiteuLike had scientific articles abstracts from varied domains like biology, physics which we didn't learn. PL

7 Process

7.1 Prior Spring Work and Motivation for Fall Semester

In last spring semester, we developed CTM (Collaborative topic Modelling [7]) as a strong baseline. Figure 12 shows the performance of collaborative topic modeling and compares it with the baseline recommendation systems when we vary the number of recommended posts. The four baseline models we have are random, popularity, LDA and topic modeling based recommendation. We see that collaborative topic modeling performs the best among the rest.

We established the following in Spring Semester: 1) Forum activities of users are not random and there is a pattern in which users seek information, 2) In order to solve the cold start problem we need to incorporate external knowledge as user forum interaction data is sparse and 3) We established the CTM has limitations. It needs good amount of past forum activities to work. CTM is a black box the topics are latent and hence no right way to find granularity and hierarchy of topics. Hence, for the Fall semester we moved to explicit concept space rather than latent topics so that we can get right level of granularity for topics through explicit concept space which is followed by learning concept hierarchies.

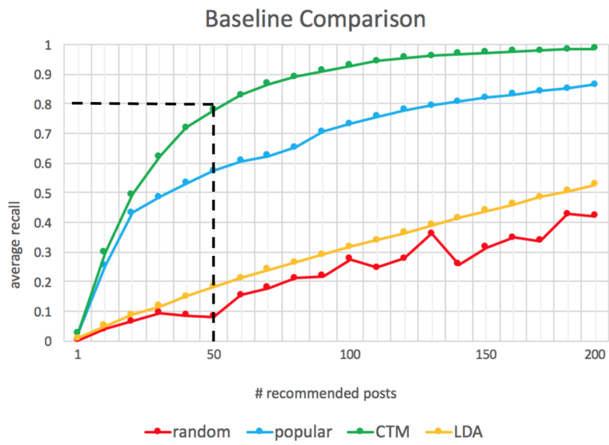


Figure 12: Baseline Comparison

7.2 Fall Development Goals

Modules	Description	Status
Data collection	- Collect data from video, wikipedia and books - Extract concepts from books and wikipedia.	Done Done
Word and Document Embeddings	- Train Word2Vec model with combined data from all three sources. - Need to create document embeddings for video transcripts. - Explore Skip Thought Vectors and SIF Model for paragraph and document level embeddings.	Done
Concept Hierarchy	- Work with the variety of distance/similarity functions to improve the hierarchy	Done
Forum Data	- Get MOOC forum data from University of Michigan*** - Evaluate on Alternative Dataset of Citeulike	Done
Recommendation Models	Concept based Forum Recommendation Model -SIF -Contextual SIF	Done
STRETCH GOALS		
Polysemy	Improve Word2Vec model using word sense disambiguation	Done

****Note: In case we don't get this data on time. we are proceeding with CiteULike dataset which holds similar assumption and satisfies the hypothesis we have put forward.*

8 Lessons Learned

Lessons Learned on Concept Hierarchy: Not all concepts can be linked into sequence of prerequisite and post-requisite concepts. Some concepts are higher level reference for a general area of research in a field like graphical models, machine learning which consists of granular concepts. When working with combined data sources (textbook/wikipedia/video) we observed that having equal representation for each concept or broad field is very important when we initially got lot of results heavily biased towards graphical models because data source was heavily biased.

Lessons Learned on Collaborating on a Data Science Project: In order to work on a large project with a team of 5 organization, modularity and documentation of code were important. Getting the right data is hard, you should always be ready with alternative sources.

9 Future Work

1. **Better Evaluation:** Get Gold labeled dataset for Concept Hierarchy and Polysemy instead of evaluating on end task.
2. **Includes a Relationship:** Incorporate "includes a relationship" into concept graph. For example, Machine learning and HMM are both concepts but there is no sequence link between them. Hence, the graph would have levels of hierarchy and will have sequence and relationship at each level of hierarchy. For example, at the most broad level "biology" and "chemistry" leads to "biochemistry".
3. **Online Update Graph:** Online model for Updating Knowledge graph given a new source. Going towards a universal concept graph for education incorporating different fields of science.
4. **Educational Search Engine:** Use the concept-graph for Educational Search Engine which generates a tutorial based on diverse sources to resolve a doubt.

10 Conclusion

We have evaluated the different components of our Concept Graph based model before testing it on the end task. We have successfully established that word2vec model captures concept similarity and different modalities/source offer different complexity of concepts which we plan to leverage. We have also shown that SIF model performs better than doc2vec and fetches highly relevant closest concepts. Finally, we attempted to test our concept hierarchy against hand labeled data of meta-academy but due to minimal intersection we could evaluate only for very few concepts. Since we didn't get forum posts Coursera data on time, we finally did our end task evaluation of our Concept-Graph based Recommendation on CiteULike dataset which is not the ideal dataset as described in Discussion s. Since both the dataset holds similar assumptions on hierarchy of concepts and user behavior. Hence, given the right dataset of learning material and forums to evaluate on we would have performed way better as we could have used concept-hierarchy and external knowledge to fullest extent.

References

[1] M. I. Lopez, J. Luna, C. Romero, and S. Ventura, "Classification via clustering for predicting final marks based on student participation in forums.," *International Educational Data Mining Society*, 2012.

[2] T.-M. Chang and W.-F. Hsiao, "Lda-based personalized document recommendation.," in *PACIS*, p. 13, 2013.

[3] Y. Cui and A. F. Wise, "Identifying content-related threads in mooc discussion forums," in *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pp. 299–303, ACM, 2015.

[4] D. Yang, M. Piergallini, I. Howley, and C. Rose, "Forum thread recommendation for massive open online courses," in *Educational Data Mining 2014*, 2014.

[5] D. Yang, D. Adamson, and C. P. Rosé, "Question recommendation with constraints for massive open online courses," in *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 49–56, ACM, 2014.

[6] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," 2016.

[7] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 448–456, ACM, 2011.

[8] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "Linear algebraic structure of word senses, with applications to polysemy," *arXiv preprint arXiv:1601.03764*, 2016.

[9] L. Pan, C. Li, J. Li, and J. Tang, "Prerequisite relation learning for concepts in moocs," 2017.

[10] MetaAcademy, *https://metacademy.org/*.

[11] Coursera, *https://www.coursera.org/specializations/data-science-python*.

[12] CiteULike, *http://www.citeulike.org/faq/data.adp*.