Facilitating Course Recommendations by Word2vec Paradigm Through Social Tags

Jingjing Wang[1], Haoran Xie[2], Oliver Tat Sheung Au[1], Lap-Kei Lee[1], Di Zou[3](✉), and Fu Lee Wang[1]

[1] School of Science and Technology, The Open University of Hong Kong, Ho Man Tin, Kowloon, Hong Kong
[2] Department of Computing and Decision Sciences, Lingnan University, Tuen Mun, New Territories, Hong Kong
[3] Department of English Language Education, The Education University of Hong Kong, Tai Po, New Territories, Hong Kong
dzou@eduhk.hk

Abstract. As an important educational form, online learning has attracted millions of registered learners, and a huge number of courses are available online. However, it is challenging for learners to identify appropriate courses from a large course pool due to the difficulties of mapping complex learning needs to the high-level course semantics. Several studies in the field of Natural Language Processing (NLP) have recently gained promising performance in capturing the semantic information. In this study, we use these NLP techniques to understand the semantics of learning needs and courses. Specifically, we model users' historical course records as word sentences using skip-gram with negative sampling to obtain course semantics. Furthermore, we introduce Laplacian Eigenmaps as the objective function and integrate the course social tags and course-user interaction as penalty factors to fine-tune the course vectors, especially the courses of different categories but similar contexts. The result verifies that the proposed method is effective for recommending suitable courses for users.

Keywords: Recommender systems · Word2vec · Word embedding · Skip-gram technique

1 Introduction

With the rapid development of the Internet and Web technologies, online learning has attracted millions of registered students worldwide (Henderikx et al. 2017). There are several reasons people tend to choose online learning. For example, learning online is cost-efficient. Another reason is that there are no restrictions on time and location for learning. Some universities or institutions also offer high-quality education free to anyone in the online learning platforms like MOOCs. The learning environment is diverse for learners worldwide (Castle and McGuire 2010). Besides, learning resources are multimedia objects, such as videos, animations, and images (Lau et al. 2013). However,

L.-K. Lee et al. (Eds.): ICTE 2020, CCIS 1302, pp. 239–247, 2020.
https://doi.org/10.1007/978-981-33-4594-2_20

confronted with a growing number of courses, it is challenging to pick up a satisfying course from the large course pool in an online learning platform as the semantics of complex learning needs and courses are difficult to understand and match. A search engine is one of the solutions for addressing this problem. In this method, course descriptions and categories are the main factors for displaying the search results. In other words, if the learner's need is obvious and can be specified by some keywords, the search engine will be sufficient for learners to identify their preferred courses. However, learners may not understand their learning needs very clear in most cases. Furthermore, online courses usually have limited text descriptions, which can be used as keyword search indices. These keywords cannot fully represent the high-level semantics of the courses. Therefore, the search engine is not a good solution when the learning needs are vague, or the courses' indices are insufficient.

The recommender systems (Resnick and Varian 1997) have been another effective solution for the above problem. From e-commerce (e.g., Amazon (Linden et al. 2003) or Alibaba) to online services (e.g., hotels (Zhang et al. 2015) or movie (Bennett and Lanning 2007)), recommender systems can assist users in clarifying their true needs and decision-making in every daily life. Recommender systems attempt to establish the linkages between users and items based on user interaction and predict users' preferences. Collaborative Filtering (CF) was one of the most popular techniques in recommender systems (Herlocker et al. 2004). In a CF-based recommender system, similar items (users) will be recommended based on the user's historical preferences. Although CF has achieved an excellent performance of recommendation, there are still some problems in this approach. A well-known issue of the CF-based approach is the cold-start problem. The system cannot understand a new user's preference as there is no or little interaction between the system and the new user.

Moreover, current course recommender systems rarely consider course contextual information in the historical user records (Ma et al. 2017; Bridges et al. 2018; Hidasi et al. 2018). The course position in the records contains semantic information and can be clustered by the semantic information. Relevant courses will share a common context in the historical user records during a period. Word2vec (Mikolov et al. 2013a, b) is one of the most commonly used Natural Language Processing (NLP) techniques that transform the unstructured natural language to the normalized and structured data with semantic information. Another issue is that learning is a spiral process of knowledge acquisition (Diamond et al. 2008). If someone tries to gain a skill, she will choose some similar courses at the first and most light-similar courses in the second step. Usually, similar courses are in the same category and have similar social tags, while the light-similar courses are not. It should be noted that the first step and the second step is not independent but alternation. However, Word2vec assumed that words share a common context will have a similar word vector. This hypothesis makes it difficult to distinguish two courses that share the same context but in different categories.

In this paper, we propose a course recommendation method by Word2vec paradigm through social tags. Firstly, we treated historical user records as the training data and used a content-based recommendation method to overcome the cold-start problem. Then we introduce Laplacian Eigenmaps as the objective function, and the course social tags and course-user interaction as the penalty factor, to fine-tune the vectors generated by the language model Word2vec. One of the advantages of the proposed method is that it can still have a stable performance for new users who have limited interactive data. The other advantage is that using penalty factors will weaken the influence of light-similar semantics and improve the recommendation's accuracy.

The remaining sections of this paper are organized as follows: Sect. 2 formalizes the problem of course recommendations. In Sect. 3, we introduce the proposed model which exploits Course2vec and integrates social tags in our proposed course recommender system. The experimental results are shown in Sect. 4. Finally, we summarize this research and discuss future directions in Sect. 5.

2 Problem Formalization

We formalize recommending courses based on historical user records and course attributes into an embedding problem. The goal is to learn $X_C \in \mathbb{R}^{|V|\times d}$, where X_C is a low-dimension vector of courses, and d is a small number of latent dimensions. For each user $u \in \mathrm{U}$, $\mathrm{U} = \left\{u_1, \cdots, u_{|U|}\right\}$, given his profile, i.e., the historical course records $Q_u = (c_1, \cdots, c_n)$ with $c_n \in C$, Q is the set of all user historical course records $Q = (Q_1, \cdots, Q_u)$ and course attribute $C = \left\{T_1, \cdots, T_{|C|}\right\}$, T_i is the course social tag. We aim to obtain the course vectors with historical user course records Q and course attribute C as formula 1.

$$X_C = f(Q, C) \tag{1}$$

3 The Proposed Model

This section gives an overview of the proposed model, then introduces a language algorithm to transform historical user records, and finally uses Laplacian Eigenmaps as the objective function to train the entire model. Our study's core contribution is that we further divide course contextual information into two types of semantic information based on the principle of knowledge acquisition. We define these two types of semantic information as follows:

Similar semantics: Courses share the same contextual information in the historical user records, and in the same category, share common social tags.

Light-similar semantics: Courses share the same contextual information in the historical user records but in different categories and have different social tags.

These two kinds of semantic information will cross-over be occurred in the user learning records. Therefore, Word2vec paradigm's significance through social tags in a recommender system include the following two aspects. Firstly, courses that have the same contextual information will locate closely in the low dimension vectors by extracting the semantic information from user behavior records. Secondly, the course vectors with social tags to distinguish similar semantics and light-similar semantics can be fine-tuned.

3.1 Generic Framework

We propose a course recommender system which exploits Word2vec to obtain semantic information (e.g., Bengio et al. 2013) from historical user records and integrate the course social tags and course-user interaction. Figure 1 illustrates the generic framework of the proposed model. The left part of Fig. 1 denotes using Word2vec to obtain the course vector, while the right one demonstrates integrating social tags to adjust the vectors.

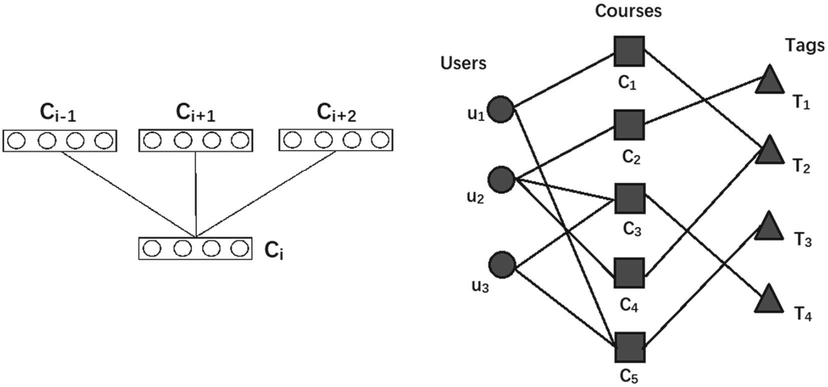

Fig. 1. The overall framework of the proposed model

3.2 Course2vec

Word2vec is most commonly used in NLP. It is a two-layer neural network model of natural language processing topics introduced by Google in 2013 (Mikolov et al. 2015). In Word2vec, sentences are regarded as an ordered sequence of words as the model input, train the network model and generate the vector to represent the word based on the assumption that words share the same context should have a close position in the target dimension. Compared to other language models like TF-IDF and Latent Semantic Analysis, these generate vectors to make the semantic information abundant (Zhao and Shang 2010) and can easily be used in the downstream tasks as word clustering and classification. User historical records can be modeled as sentences in Word2vec because they are both generated by basic elements based on semantic rules. Words co-occurred in the sentence contain semantic information, while relevant courses are located closely in the records during a period.

Word2vec contains two sub-models which are the CBOW model and the Skip-gram model. These two models can be used in different scenarios. In the CBOW model, surrounding words are the input; the model estimates the center words' likelihood. While in the Skip-gram model, center words are the input, the model aims to predict its neighbors. As our goal is to predict the next selected item based on the existing records, this study employs the Skip-gram model.

Similar to Word2vec, we propose the Course2vec to model the course representation. Given the set of courses $Q_u = (c_1, \cdots, c_n)$ taken by one user. Sequences

$Q = (Q_1, \cdots, Q_u)$ are the ordered course list. We aim to maximize the probability of the contextual words in the sequence.

$$y_t = \frac{1}{T}\sum\nolimits_{t=1}^{T}\left(\sum\nolimits_{-c\leq j\leq c, j\neq 0}\log p\left(c_{t+j}|c_t\right)\right) \tag{2}$$

where $c_1, \ldots, c_T$ are courses in the training corpus, and c is the length of the window around target course c_t. Train the network by feeding it word pairs $<c_{t+j}, c_t>$. The probability $\Pr\left(c_{t+j}|c_t\right)$, the important part of the objective y_t, is given by SoftMax:

$$\Pr\left(c_{t+j}|c_t\right) = \frac{\sum e^{v_{c_{t+j}}.v_{c_t}}}{\sum_{c'\in C} e^{v'_c.v_{c_t}}} \tag{3}$$

where v_{c_t} and $v_{c_{t+j}} \in \mathbb{R}^d$ are vector representations for course c_t and c_{t+j}, respectively. Training the whole neural network indicates maximizing the function $\Pr\left(c_{t+j}|c_t\right)$. In other words, each training $\Pr\left(c_{t+j}|c_t\right)$ will adjust all the neural network cells, which will slow down the training process. Negative sampling addresses this problem by only modify a small number of negative courses to update the weights. The objective function can be defined as follows:

$$J = log\sigma(c_w.c_t) + \sum\nolimits_{i=1}^{k} log\sigma\left(-c_{N,i}.c_t\right) \tag{4}$$

where σ denotes the sigmoid function, and k is the total number of randomly selected negative courses.

3.3 Laplacian Eigenmaps

Although the users' historical records contain contextual information, language models can transform contextual information into vectors. There are two types of semantic information: *similar semantics* and the other is the *light-similar semantics*. To deal with this issue, we propose to use Laplacian Eigenmaps as the objective function to generate the low-dimensional representation and fine-tune the course vectors obtained by Course2vec, especially courses share common context but in different categories. The objective function is defined as follows:

$$\mathcal{L} = \mu_{ij}\left|\left|y_i - y_j\right|\right|_2^2 \tag{5}$$

where y_i, y_j is the final vector of courses i, j, μ_{ij} is the weight between them. In the model, μ_{ij} is a penalty factor, if there is a slight penalty, course i and course j will locate far in the final space. We aim to minimize the objective function to ensure that y_i and y_j are close if course i and course j have a heavy penalty.

$$argmin\,\mathcal{L} = \mu_{ij}\left|\left|y_i - y_j\right|\right|_2^2 \tag{6}$$

In our model, there are two factors related to μ_{ij}. The first one is the number of common interactive users, and the second one is the number of social tags, μ_{ij} can be defined as follows.

$$\mu_{ij} = \frac{U_i \cap U_j}{U_i \cup U_j} + \frac{T_i \cap T_j}{T_i \cup T_j} \tag{7}$$

where U_i is the set of user enroll course i and U_j is the set of user enroll course j, T_i is the set of social tags of the course i, and T_j is the set of course j social tags. These social tags are obtained by using Text-CNN (Kim 2014) from analyzing the course introduction and user online comments. To solve our model's objective function, we regard the course as the node in the graph. Given a network $G = (V, E)$, in which V and E are the set of course node and edge, respectively. We can obtain course adjacency matrix S. For each instance in S, $w_i = \left\{\mu_{ij}\right\}_{j=1}^{n}$, $\mu_{i,j}$ means the links between c_i and c_j. Therefore, the objective function can be rephrased as follows:

$$argmin\, \mathcal{L} = \mu_{ij}\left|\left|y_i - y_j\right|\right|_2^2 = 2tr\left(Y^T LY\right) \tag{8}$$

where $L = D - S$, D is the diagonal matrix, and $D_{i,i} = \sum_j s_{i,j}$, Y is the prediction vector. Finally, the model generates the final representation of the course with the user's historical behavior information. In this paper, cosine similarity is employed for measuring the courses' similarity with the low-dimensional vectors. At last, *top-N* most similar courses will be recommended to the user based on the course similarity.

4 Experiment

In this paper, the dataset was introduced in the section of MOOC data selected from XuetangX. XuetangX, launched in October 2012, provided over 1,000 courses distributed 12 categories from art, biology to computer science. More than 10,000,000 users have registered in XuetangX. In this experiment, we selected enrolled behaviors from October 1st, 2016 to December 30th, 2017, as samples. Each instance in the training set was a sequence of historical records. For each course, we constructed social tags by using text-CNN to extract from the course introduction. In this experiment, we selected the *top-5* social tags as the course tags. Table 1 showed the social tags of some courses.

Table 1. The social tags of sample courses

Course	*Social tags*
Java Programming: Solving Problems with Software	Programming, Duke university, Problem-solving skills, Java program, Harder
Introduction to Computer Science and Programming	Computer Graphics, Mathematic, JavaScript, Computer Programming, Beginner
Mathematics for Machine Learning	Machine Learning, Beginner, Data Science, Python Programming, Gradient Descent

Firstly, we sorted the user-selected historical records during the period as the dataset. Then, we built a Course2vec model using python and genism to transform the semantic information to the word representation. Finally, to avoid courses in different categories share common context in Course2vec, we used the Laplacian Eigenmaps method as the objective function to fun-tune the Course2vec vectors. The penalty factor in the objective function was based on user-item-tag interaction.

Metrics and Baselines. We compared our proposed method with the following two methods measured by the two metrics: Recall@20 and Precision@20.

Collaborative Filtering (CF): This method regarded the user-item interactions as the original user vector. If the user enrolled in the course, the interactive matrix's value equals 1, else 0.

Matrix Factorization (MF): This method modeled user preferences by decomposing the user-item interactive matrix to obtain user and item embeddings.

Parameter Settings. To gain the course vectors by Course2vec, we constructed the model with a 64-dimensional course vector as output, selecting five neighbors as the context, five courses as the negative, and neglecting the course that had been occurred less than five times. Here we chose a relative high dimension vector because high dimension reserves rich information, and in the following step, Laplacian Eigenmaps reduced the number of vector dimensions to 32. According to each course's selected records, we obtained the similarity of each course by the course vector and recommend *top-N* courses.

Table 2. Performance of course recommendations by using different models

Method	*Recall@20*	*Precision@20*
CF	32.10%	21.96%
MF	35.30%	22.50%
Course2vec	35.70%	22.70%
Course2vec ++	35.80%	23.00%

Table 2 showed the results of course recommendations in terms of Recall@20 and Precision@20. Course2vec ++ was a model of the Course2vec with course social tags and course-user interactions. The proposed Course2vec and Course2vec ++ models outperformed all baselines. The second best performance achieved by Course2vec also verified that incorporating semantic information can improve the effectiveness of the recommendation. Course2vec ++ achieved the best performance and was lightly better than Course2vec. Table 3 showed a qualitative example of recommendation results for a specific user (i.e., the user id is 58 in the dataset) by using two different models (i.e., Course2vec and Course2vec ++).

Table 3. A qualitative example of recommendation results of a specific user (id 58)

Course name	*Recommend Results*	
	Course2vec	Course2vec ++
Advanced Machine Learning	Mathematics for Machine Learning	Mathematics for Machine Learning
	Machine Learning with TensorFlow	Machine Learning with TensorFlow
	Python 3 Programming	Data Science: Statistics and Machine Learning
	Data Science: Statistics and Machine Learning	Deep Learning
	Python for Data Science and AI	Machine Learning with Python

5 Conclusion

This paper proposed a method of course recommendation. First, we represented the user's historical records as word sentences through exploiting the skip-gram with negative sampling to obtain course embeddings. To avoid two types of semantic information that share common context in the records, we integrated the course social tags and course-user interactions as penalty factors to adjust course embeddings under different categories. Experimental results showed that Course2vec captured semantic information from the records. Compared to recommendations based on Course2vec, Course2vec + + improved the recommendation accuracy. However, this paper focused on observed interactions between users and courses from the perspective of their bipartite graph. For our future research, we plan to integrate information extracted from unobserved items to improve the recommendations' accuracy and serendipity.

Acknowledgement. The work described in this paper was supported by the Katie Shu Sui Pui Charitable Trust — Academic Publication Fellowship (Project Reference No. KSPF2019-03).

References

Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)

Castle, S.R., McGuire, C.J.: An analysis of student self-assessment of online, blended, and face-to-face learning environments: Implications for sustainable education delivery. Int. Educ. Stud. **3**(3), 36–40 (2010)

Diamond, N., Koernig, S.K., Iqbal, Z.: Uniting active and deep learning to teach problem-solving skills: Strategic tools and the learning spiral. J. Mark. Educ. **30**(2), 116–129 (2008)

Henderikx, M.A., Kreijns, K., Kalz, M.: Refining success and dropout in massive open online courses based on the intention–behavior gap. Distance Educ. **38**(3), 353–368 (2017)

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. (TOIS) **22**(1), 5–53 (2004)

Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)

Lau, R.W.H., Yen, N.Y., Li, F., Wah, B.: Recent development in multimedia e-learning technologies. World Wide Web **17**(2), 189–198 (2013). https://doi.org/10.1007/s11280-013-0206-8

Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013a)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013b)

Mikolov, T., Chen, K., Corrado, G.S., Dean, J.A.: U.S. Patent No. 9,037,464. U.S. Patent and Trademark Office, Washington, DC (2015)

Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3), 56–58 (1997)

Bridges, C., Jared, J., Weissmann, J., Montanez-Garay, A., Spencer, J., Brinton, C.G.: Course recommendation as graphical analysis. In: 2018 52nd Annual Conference on Information Sciences and Systems (CISS), pp. 1–6. IEEE (March 2018)

Bennett, J., Lanning, S.: The netflix prize. In: Proceedings of KDD cup and workshop, vol. 2007, p. 35 (August 2007)

Zhao, Z.D., Shang, M.S.: User-based collaborative-filtering recommendation algorithms on hadoop. In: 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 478–481. IEEE (January 2010)

Zhang, K., Wang, K., Wang, X., Jin, C., Zhou, A.: Hotel recommendation based on user preference analysis. In: 2015 31st IEEE International Conference on Data Engineering Workshops, pp. 134–138. IEEE (April 2015)

Ma, H., Wang, X., Hou, J., Lu, Y.: Course recommendation based on semantic similarity analysis. In: 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), pp. 638–641. IEEE (August 2017)

Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 843–852 (October 2018)