International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

A Course Knowledge Analysis Framework for On-line Education

Mingxi Zhang[1], Jianghai Dai[2], Yuqing Su[3], Dini Xu[4]

[1, 2, 3, 4]College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai, China
[1]Corresponding author E-Mail: *WAXL7461[at]aliyun.com*

Abstract: ***Online education has been widely welcomed by learners for its convenience and rich interactivity. However, when facing the huge number of online courses, learners will have difficulty in choosing. The knowledge points contained in the course can effectively reflect the main teaching tasks of the course, making it easy for learners to quickly select courses they need. In this paper, we propose a random walk-based system for Courses knowledge analysis in a tag-knowledge bipartite network. First, we use TextRank to extract keywords from course texts as tags to describe the knowledge points according to annotated data. Next, tag-knowledge bipartite network is constructed by using the tags and knowledge points as nodes and the descriptive relationships between them as edges. Finally, we use Random walk to measure the relevance score between courses and knowledge points then return the top k relevant knowledge points. Experiments on real data sets have demonstrated the effectiveness and accuracy of the system.***

Keywords: course knowledge; random walk; bipartite network; relevance score

1. Introduction

With the rapid development of information and communication technology, the online learning [1] platforms represented by MOOC and Coursera have been well known by learners. The high quality and variety of online courses make online learning an important way for learners to learn autonomously. However, the large number and variety of online courses cause the problem of information overload and make course selection more difficult for learners. Course knowledge points can effectively reflect main teaching tasks of the course which can help the online learning platform to manage courses and facilitate the course selection for learners. Traditional knowledge points analysis relies on manual annotation, which is costly and does not guarantee the quality. Therefore, automating the knowledge points analysis is the problem that needs to be solved.

The goal of knowledge points analysis is to analyze the knowledge composition of the course using information available in the course such as text. And it has many applications in the field of education, such as course recommendation [2], course management [3], knowledge tracing [4], and cognitive diagnosis [5].

Existing methods for analyzing course knowledge points are broadly divided into two categories, namely content-based methods and link-based methods [6]. Content-based methods use information extraction techniques [7] in natural language processing [8] to extract knowledge points which belong to the supervised learning. Link-based methods extract knowledge points from text by building a word graph and using ranking algorithms to analyse the graph.

The difficulty of the knowledge points analysis is grasping the relationship between knowledge points accurately. Tag-knowledge bipartite network provides an idea to express the relationship between tags from text and knowledge points by regarding them as nodes and relationship between them as edges. However, this simple approach can hardly capture the potential association between nodes. Based on such a problem, we introduce Random walk [9] to solve. With its advantage, it has been widely used in stochastic models [10] and graph representation learning [11].

In this paper, we propose a Random walk-based system for analyzing course knowledge points in a tag-knowledge bipartite network and the framework is shown in Figure 1. Firstly, we have to preprocess the texts from courses, including word segmentation, stop word removal, word stemming and keyword extraction. Next, keywords are considered as knowledge tags along with knowledge points as nodes on the graph and the descriptive relationships between them are regarded as edges to build a tag-knowledge bipartite network. Then, considering the different ability of different tags to describe knowledge points, we weight the edges using TF-IDF [12] and remove the weak links with low weights which could be noisy. Finally, we use Random walk to calculate the relevance score between the tags and knowledge points and perform weighted-sum for a given course text and return the top k related knowledge points. Our main contributions are as follows:

1) Based on random walk model, we proposed a system for courses knowledge analysis. And the keywords or tags extracted from courses were used as the basis of knowledge points analysis.
2) Based on the relationships between the tags of the course and the knowledge points, we constructed a tag-knowledge bipartite network. When analyzing knowledge points of course, our system tends to search knowledge points that have similar knowledge description relationships.
3) We conducted experiments on a real dataset and found that for a given course text, the system can return knowledge points related with high confidence, proving the validity and accuracy of our proposed system.

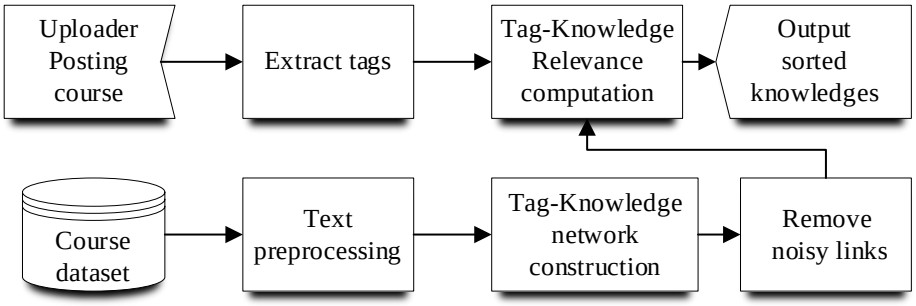

Figure 1: System framework

2. Related Work

Courses knowledge analysis is broadly classified into two categories, which are content-based methods and link-based methods. Content-based methods use supervised learning and label knowledge points for words in partial texts then train a word classifier. Link-based methods use texts to construct a word graph then extract knowledge points in an unsupervised manner and rank after that.

Content-based methods learn knowledge points with supervised methods and train a word classifier, which could be a binary or multi-categorical classification task. In the case of binary classification, it only requires determining whether the word represents a knowledge or not, while in the case of multi-categorical classification, it requires determining to which specific knowledge the word belongs. Depending on the classifier, naive bayes [13], decision tree [14], support vector machine [15] or XGBoost [16] can be used.

Link-based methods belong to unsupervised learning. TextRank [17] constructs a word graph and extracts keywords by ranking words based on co-occurrence relationships in text, ExpandRank [18] uses neighboring documents to construct a word graph to enhance keyword extraction, and Phraseformer [19] uses graph embedding for keyword extraction.

The content-based methods fail to capture the potential relationships between words when classifying. And the link-based methods extract keywords as knowledge points by default, ignoring the case that knowledge points are more abstract and do not appear directly in the text.

3. Network Construction

3.1 Data preprocessing

The primary goal of data preprocessing is to extract keywords from the input texts of courses and regard them as tag nodes in tag-knowledge bipartite network. When processing texts of courses, operations such as sentence splitting, word segmentation, stop words removal, punctuation removal, special character removal, word stemming, and word case normalization are required. Sentence splitting and word segmentation are used to split long text into collections of words. Stop words, punctuation, and special character removal are used to reduce redundant information in the word collections that are not related to the main content of the text. Word stemming and case normalization are used to reduce the size of the nodes in tag-knowledge network to be built and to save computational overhead. Specifically, since the text used in the experimental dataset contains both Chinese words and English words, we choose Jieba which can specify custom dictionary and stop words dictionary to segment words and remove stop words. Next, for the sliced English words, we extracted stems using Porter Stemmer [20] and then case-normalized the words.

After obtaining a clean list of words, we use TextRank to perform keyword extraction for each text, which can be used to describe the key messages of the course. Then we add all the keywords to the offline tag warehouse which tag-knowledge network can use for relevance computation. In the online step, when a new course is published, the keywords of the course text are obtained in the same way as the tag nodes of the tag-knowledge network, and the relevance score between tags and knowledge points is computed based on the offline results.

TextRank is used to extract key information from text, such as keywords, text summaries, etc., and has a wide range of applications in information retrieval and data mining as well as natural language processing. The idea is derived from the web ranking algorithm PageRank [21], which obtains the required key information by iterative computation. In this paper, TextRank is used to obtain the keywords in the course text. Specifically, after obtaining a clean word list, a directed graph $G' = (V', E')$ is constructed based on the word order relationship, where V' denotes the set of nodes constituted by the words, E' denotes the set of directed edges between nodes. The importance of all the nodes in the word graph is obtained as

$$S(v_i) = (1-d) + d \times \sum_{j \in I(v_i)} \frac{w_{ij}}{\sum_{v_{k'} \in O(v_j)} w_{jk'}} S(v_j) \#(1)$$

where w_{ij} denotes the directed edge weights of node $v_i \in V'$ pointing to node $v_j \in V'$. $I(v_i)$ denotes the set of nodes pointing to v_i, and $O(v_j)$ denotes the set of nodes pointed by v_j. $d \in [0,1]$ is the damping factor, which generally takes the value of 0.85. After obtaining the scores of all nodes and sorting them in descending order, the top k keywords can be selected.

3.2 Tag-Knowledge network construction

Tag-knowledge network is defined as a bipartite graph $G = (V, E)$, $V = V_t \cup V_k$, V_t and V_k are the set of tags and knowledge points, respectively. E denotes the set of edges consisting of the description relations of tags to knowledge points, and the edge $e(k_i, t_j) \in E$ denotes the knowledge $k_i \in V_k$ can be described by the tag $t_j \in V_t$.

For a given course text, we can obtain a number of keywords as tags, which form a tag-knowledge bipartite network with uniquely identifiable knowledge. For a course, it can be represented by several tags, and each tag can be regarded as a description of multiple knowledge points. Similarly, for a knowledge, it can be described by multiple tags, which is a many-to-many relationship in general. In the process of courses knowledge analysis, we need to get the tag set of course text and use it as multiple queries to find the knowledge points related to the course itself. Figure 2 shows a tag-Knowledge network.

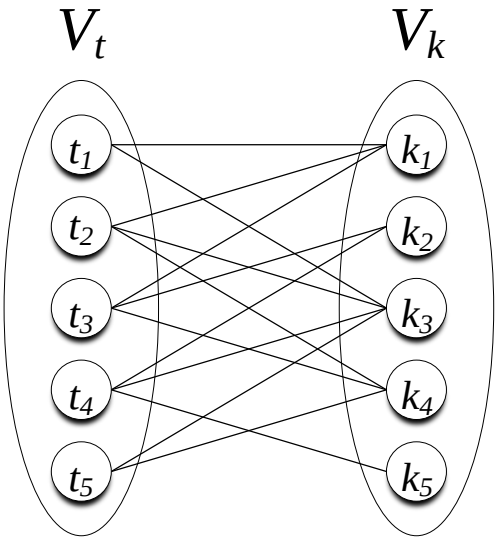

Figure 2: Tag-Knowledge bipartite network

3.3 Weight optimization

For certain knowledge, it can be described by multiple tags, however, different tags may have different descriptive ability for knowledge, so tags need to be treated differently. In addition, some tags with low descriptive ability will affect the result of courses knowledge analysis and bring extra computational overhead, so pruning strategy be introduced in our system to reduce the impact comes from noisy link.

TF-IDF is a statistical method that is used to distinguish the importance of different words to a document. The idea can be simply summarized as follows: if a word is important to the current document, it should satisfy two conditions. first, it appears more often in the current document, and second, it appears less often in all documents. Inspired by the TF-IDF idea, in the tag-knowledge bipartite network, knowledge points and tags can be analogous to words and documents respectively, so it can be used to model the descriptive ability of a tag to a knowledge and we add the descriptive ability numerically as weight to the tag-knowledge bipartite network. Specifically, in the scenario of courses knowledge analysis, the TF-IDF involves two parts of computation, the tag frequency TF and the inverse tag frequency IDF, and finally the product of them is used as the importance of a tag to a knowledge. The specific calculation is shown as

$$\mathrm{TF}_{t,k} = \frac{n_{t,k}}{|O(k)|} \#(2)$$

$$\mathrm{IDF}_t = log\frac{|K|}{(|Q|+1)} \#(3)$$

$$\mathrm{TFIDF}_{t,k} = \mathrm{TF}_{t,k} * \mathrm{TF}_{t,k} \#(4)$$

In Equation (2), $\mathrm{TF}_{t,k}$ denotes the frequency of tag t appearing in knowledge k as description, $n_{t,k}$ denotes the number of times tag t appears in knowledge point k as description, and since the tag in the text appears only one time in the description of k, $n_{t,k}$ is constant to 1, $|O(k)|$ denotes the number of tags owned by knowledge k. In Equation (3), IDF_t denotes the inverse tag frequency of tag t, $|K|$ denotes the number of non-repetitive knowledge points, $|Q|$ denotes the number of knowledge points possessing tag t, and 1 is the smoothing factor which ensures denominator is not equal to zero. In Equation (4), $\mathrm{TFIDF}_{t,k}$ indicates the importance of tag t to knowledge$k$.

Courses knowledge analysis relies on Random walk to calculate the relevance score between tags and knowledge points. For a sparse network, Random walk is able to capture the potential association between nodes using global information. Besides, Random walk uses iterative computation to obtain relevance score between nodes, and it is proved that it can converge after multiple iterations. For a network with n nodes, the iterative method is shown as

$$\overrightarrow{q_x^l} = (1-\alpha)P^T\overrightarrow{q_x^{l-1}} + \alpha\overrightarrow{e_x} \#(5)$$

where l denotes the iteration step, $\overrightarrow{q_x^l} \in \mathbb{R}^{n\times 1}$ denotes the iteration result of the current step, $\overrightarrow{q_x^{l-1}} \in \mathbb{R}^{n\times 1}$ denotes the iteration result of the previous step, and $\overrightarrow{e_x} \in \mathbb{R}^{n\times 1}$ denotes the vector of query nodes. In this vector, only the value at the query node index is 1 and the rest are 0.$\overrightarrow{e_x} \in \mathbb{R}^{n\times 1}$ denotes the normalized transfer probability matrix, i.e., each row sums to 1. The weight w_{ij} denotes the transfer probability from node v_i to node v_j, $P^T \in \mathbb{R}^{n\times n}$ is the transpose matrix of P. α is the restart probability, indicating that during the random walk process, the current node has a probability of α jumping to the start node and restarting the random walk process, and a probability of $1-\alpha$ to select the neighbor node as the next jump. The vector obtained after convergence of the algorithm indicates the relevance score of the query node and all nodes in the graph, the larger the score, the more relevant the node is to the query node. In the scenario of courses knowledge analysis, all tags from a course text are used as query nodes for relevance computation, and after convergence of iterations, the relevance score between the course and all knowledge points are obtained by weighted summation of each query result vector, from where top k knowledge points are selected as the final result. The relevance score between courses and knowledge points can be calculated as

$$S_{ck} = \sum_{t\in tags(c)} R_{tk} I(t) \#(6)$$

where S_{ck} denotes the relevance score of course c to knowledge k, $t \in tags(c)$ denotes the tags or keywords of course c, R_{tp} denotes the relevance score between tag$t$ and knowledge k, $I(t)$ denotes the importance of keyword t computed by TextRank, i.e., different keywords of the course have different query weights.

4. Experiments

4.1 Dataset

The dataset used in this paper is from [22]; contains 706

courses, 38, 181 videos, 114, 563 knowledge points, 167, 751 mapping information between courses and knowledge points, and 31, 948 mapping information between videos and knowledge points. The text data of courses are obtained from the course introduction and the instructor's speech of the videos. Finally, in tag-knowledge bipartite network, we get 116, 661 tags, 114, 527 knowledge points, and 780, 318 descriptive relationships between tags and knowledge points.

4.2 Experimental environment

The experiments run on Intel (R) Xeon (R) Bronze 3106 CPU[at]1.70GHz and 128GB RAM, under Windows 2012R. The development environment is VSCode 2019. We use Python 3.8.7 to process the dataset and Java 11 to implement the algorithms.

4.3 Result

In order to verify the effectiveness of our system, three randomly selected excerpts of the input text were used to demonstrate the effect, and both the input text and the output text were translated into English.

Text of Course 1: In 1895, when Roentgen was doing this experiment with a cathode ray tube, he found that a place outside the cathode ray tube would glow faintly. He repeated the experiment and found that this was not a cathode ray, because cathode rays are electron beams, which are charged. . .

Text of Course 2: When comparing two means for quantitative data, again there are two methods. One is the confidence interval method and the other is the significance test method. Confidence intervals are probably more useful than significance tests, so we will now focus on the confidence interval method. . .

Text of Course 3: A public fundraising project is an investment in the initiator's dream of public welfare, with elements of innovation, happiness and positive energy. However, it requires a higher quality of service from the platform, and requires the establishment of a social center with the initiator as the core, and the continuous spread of fundraising through interpersonal communication. . .

Table 1 shows the top-5 results returned for the texts of Courses 1-3. Course 1 contains terms such as ray and electron beam, which can intuitively feel that the course is related to physics knowledge points, and the returned results of positronium, radioisotope, antiparticles, metastable stat, and paradox all meet the requirements. Through a similar analysis method, it is concluded that Course 2 introduces the knowledge points related to probability theory, and the returned results also meet the requirements. Course 3 introduces the knowledge points of management such as crowdfunding and financing, and the returned results meet the requirements. The analysis of several sets of experimental results verifies the effectiveness and accuracy of the system.

Table 1: Result of courses knowledge analysis

Rank	Course 1	Course 2	Course 3
1	电子偶素 (positronium)	随机变量 (randomvariable)	竞争优势 (competitiveadvantage)
2	放射性同位素 (radioisotope)	概率分布 (probabilitydistribution)	创业计划书 (businessplan)
3	反粒子 (antiparticles)	条件概率 (conditionalprobability)	竞争分析 (competitionanalysis)
4	亚稳态 (metastablestat)	贝叶斯公式 (bayesformula)	融资方式 (financingmethods)
5	佯谬 (paradox)	联合分布 (jointdistribution)	资本结构 (capitalstructure)

5. Conclusion

In this paper, we propose a Random walk-based system for courses knowledge analysis in the tag-knowledge bipartite network, which is divided into two parts. In the first part, the course text keywords or tags and knowledge points are regarded as nodes, the descriptive relationships between them are regarded as edges to construct tag-knowledge bipartite network. In the second part, the course text tags are used as queries, and the relevance score of the course and the knowledge is computed on the tag-knowledge network using Random walk, and the top k relevant knowledge points are returned according to the relevance score. An empirical study on a real dataset has demonstrated the effectiveness and accuracy of the system. In the future work, we will improve the system's adaptability and stability to large-scale networks from the perspective of computational efficiency.

6. Acknowledgment

This work was supported by National Natural Science Foundation of China under Grant 62002225, and Natural Science Foundation of Shanghai under Grant 21ZR1445400.

References

[1] Muilenburg, L. Y., & Berge, Z. L. (2005). Student barriers to online learning: A factor analytic study. Distance education, 26 (1), 29-48.
[2] Zhang, H., Huang, T., Lv, Z., Liu, S., & Zhou, Z. (2018). MCRS: A course recommendation system for MOOCs. Multimedia Tools and Applications, 77 (6), 7051-7069.
[3] Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system. In EdMedia+ Innovate Learning (pp.171-178). Association for the Advancement of Computing in Education (AACE).
[4] Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4 (4), 253-278.
[5] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. Proceedings of the national academy of sciences, 111 (23), 8410-8415.

[6] Borodin, A., Roberts, G. O., Rosenthal, J. S., & Tsaparas, P. (2005). Link analysis ranking: algorithms, theory, and experiments. ACM Transactions on Internet Technology (TOIT), 5 (1), 231-297.
[7] Jiang, J. (2012). Information extraction from text. In Mining text data (pp.11-41). Springer, Boston, MA.
[8] Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37 (1), 51-89.
[9] Tong, H., Faloutsos, C., & Pan, J. Y. (2006, December). Fast random walk with restart and its applications. In Sixth international conference on data mining (ICDM'06) (pp.613-622). IEEE.
[10] Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., & Upfal, E. (2000, November). Stochastic models for the web graph. In Proceedings 41st Annual Symposium on Foundations of Computer Science (pp.57-65). IEEE.
[11] Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78-94.
[12] Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Systems with Applications, 38 (3), 2758-2765.
[13] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol.3, No.22, pp.41-46).
[14] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21 (3), 660-674.
[15] Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24 (12), 1565-1567.
[16] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp.785-794).
[17] Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp.404-411).
[18] Wan, X., & Xiao, J. (2008, July). Single Document Keyphrase Extraction Using Neighborhood Knowledge. In AAAI (Vol.8, pp.855-860).
[19] Nikzad-Khasmakhi, N., Feizi-Derakhshi, M. R., Asgari-Chenaghlu, M., Balafar, M. A., Feizi-Derakhshi, A. R., Rahkar-Farshi, T.,. . . & Ranjbar-Khadivi, M. (2021). Phraseformer: Multimodal Key-phrase Extraction using Transformer and Graph Embedding. arXiv preprint arXiv: 2106.04939.
[20] Karaa, W. B. A., & Gribâa, N. (2013). Information retrieval with porter stemmer: a new version for English. In Advances in computational science, engineering and information technology (pp.243-254). Springer, Heidelberg.
[21] Langville, A. N., & Meyer, C. D. (2004). Deeper inside pagerank. Internet Mathematics, 1 (3), 335-380. Yu, J., Luo, G., Xiao, T., Zhong, Q., Wang, Y., Feng, W.,. . . & Tang, J. (2020, July). MOOCCube: a large-scale data repository for NLP applications in MOOCs. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp.3135-3142).
[22] Yu, J., Luo, G., Xiao, T., Zhong, Q., Wang, Y., Feng, W.,. . . & Tang, J. (2020, July). MOOCCube: a large-scale data repository for NLP applications in MOOCs. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp.3135-3142).