2018 IEEE Third International Conference on Data Science in Cyberspace

Course Concept Extraction in MOOC via Explicit/Implicit Representation

Xiaochen Wang†, Wenzheng Feng†, Jie Tang†, Qiangyang Zhong†,
† Department of Computer Science and Technology, Tsinghua University, Beijing, China
{xiaochen15, fwz17, zhongqy16}@mails.tsinghua.edu.cn, {jietang}@tsinghua.edu.cn

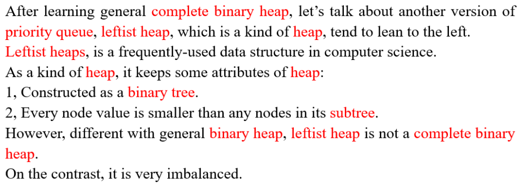
***Abstract*—Massive Open Online Courses(MOOCs) provide convenient access to knowledge for learners all over the world. Concept Extraction is a basic requirement in MOOCs. However, textual content in MOOCs, such as video subtitles and quizzes, are generally presented as semi-structured or unstructured format. Thus it is hard to extract important concepts with simple methods from MOOCs. In this paper, we design a graph-based propagation method to solve the concept extraction problem. Our method utilize textual and structured data on Wikipedia, to generate implicit and explicit representation for concepts respectively. Experiments show that our method outperforms alternative methods on Chinese dataset(+0.054-0.062 in terms of MAP).**

***Index Terms*—Natural language processing, Knowledge representation, Concept extraction, MOOC**

I. Introduction

The rapid development of Massive open online courses (MOOCs) has significantly enriched our learning channels. Now students can use computers, phones to take online courses from prestigious universities remain within doors. Following this trend, MOOC platforms have been developed quickly around the world and the number of courses on it are growing rapidly over year. As a result, large volumes of education resources especially course videos and its subtitles are created. One fundamental analytic operation is to recognize key concepts in documents which is particularly necessary under MOOC circumstances. In course video subtitles, concepts are usually referred to as subject terms in that course(e.g. Binary Search Tree in course Data Structure). Fig. 1 shows an example of course concepts. Extracting concepts from course text can not only help students better grasp the main points in the video, but also liberate teachers from heavy work of human labeling. From research perspective, automatical concept extraction in MOOCs supports research towards deeper analysis such as video content summary and MOOC knowledge graph construction.

Course concept extraction is non-trivial for two reasons. The first reason is the low-frequency problem. Course video captions often involve many low frequency course concepts, primarily because these concepts are from other courses thus not discussed in detail. Statistical methods such as TF-IDF [1] and C-value/NC-value [2] often disregard informative words with low frequency, thus may produce low coverage of course concepts. Because when concept frequency is low, statistical methods may fail to provide reliable estimates of their statistical quantity (e.g., IDF). Meanwhile, these low-frequency concepts are usually the most confused concepts for learners. Because they are not explained clearly in the course. The second reason is short context problem. Comparing with documents of certain domain like news and academic papers, course video captions have shorter context but more concepts. As course video subtitles provide insufficient context information, context-based methods such as co-occurrence [3] and inter-domain entropy [4] would achieve unsatisfactory results.



After learning general complete binary heap, let's talk about another version of priority queue, leftist heap, which is a kind of heap, tend to lean to the left.
Leftist heaps, is a frequently-used data structure in computer science.
As a kind of heap, it keeps some attributes of heap:
1, Constructed as a binary tree.
2, Every node value is smaller than any nodes in its subtree.
However, different with general binary heap, leftist heap is not a complete binary heap.
On the contrast, it is very imbalanced.

Fig. 1. Example of a MOOC script and its concepts(colored phrases).

Therefore, it is hard to extract concepts with only course textual content. External knowledge must be involved to solve above problems. Wikipedia, as the most famous online encyclopedia, is frequently used to generate knowledge base [5]. In our work, we utilize both structured and unstructured data in Wikipedia to provide external knowledge for concept extraction.

Concepts in courses has semantic relations, it is critical to explore the relationships between them to make concept extraction. Our extraction method start with an intuitionistic idea: if two concepts has close semantic relations, and one of them are key concepts in a certain course, then the other one is likely to be a concept of that course. We use candidate concepts as vertexes to construct a graph. Edges in the graph are weighted by the semantic similarity of concepts on both ends of the edge. Our method start from some known concepts, they spread "concept score" on the graph. After iterations, candidate concepts with high scores are the course concepts that we want.

In the rest of this paper, we will first introduce some related works, then define the problem and describe our method.

978-1-5386-4210-8/18/$31.00 ©2018 IEEE
DOI 10.1109/DSC.2018.00055

Some experiments are made to verify the effectiveness of our method. And we also analyze the details in the experimental results.

II. Related Works

The related works are mainly about keyphrase extraction, which focus on extracting important and topical phrases from one document automatically [14]. Generally speaking, keyphrase extraction methods can be categorized into two groups: supervised methods and unsupervised methods, and we review these two types of approaches in this section.

As for supervised methods, the task is formalized as a binary classification problem: each phrase in document is input a classifier to learn a probability indicating whether the phrase is a keyphrase or not [15]. Different classifiers are used in previous work, such as naive bayes [16], [17],maximum entropy [19], [20], support vector machines [20], [21] and decision trees [14]. For unsupervised methods, each candidate phrase is assigned a saliency score based on some related features [22]. In general, document information, including *tf-idf*, co-occurrence or neighbor documents are frequently considered in this scenario. For example, ExpandRank [22] utilize the set of neighborhood documents to enhance keyphrase extraction in single document. TextRank [9] employ word co-occurance graph for ranking keywords. Huang et al. [23] construct a word semantic network for each document based on co-occurance information and capture the importance of each phrase by analyzing the network. Liu et al. [24] propose a novel method which integrates phrasal segmentation into phrase extraction framework.

Although substantial methods are proposed to address this task, extracting low-frequency keyphrases from a document still remains a difficult challenge. To tackle this problem, some work employ the external knowledge together with the document information to improve the performance of automatic keyphrase extraction. For example, KEA++ system [18] obtains the candidate phrases from a domain-specific thesaurus. Gazendam et al. [25] also use the thesaurus as background corpus. Apart from thesaurus, some work also employ knowledge bases to calculate semantic relations between concepts. Vivaldi et al. [27] consider the an ontology hierarchy acquired from Wikipedia categories as the external knowledge. Similarly, Berend et al. [28] utilize the features from Wikipedia level to achieve enhancements for performance. Rospocher et al. [26] use WordNet to detect synonym concepts, and rank the candidate concepts according to the result. All these methods make use of the explicit knowledge contained from external source. Different from them, we solve this problem via both explicit knowledge and implicit representations of knowledge.

III. Problem Definition and Framework

In this section, we first give some necessary definitions and then formulate the problem of our work.

Course Corpus is composed by n courses, denoted as $\mathcal{D} = \{\mathcal{C}_j\}_{j=1,\ldots,n}$, where $\mathcal{C}_j$ is one course. Course $\mathcal{C}_j = \{v_{ij}\}_{i=1,\cdots,m_j}$ consists of m_j course videos, where v_{ij} stands for the i-th video. Each video v_{ij} is composed of its video texts (video subtitles or speech script) $\{d_i\}$, each d_i is a single word in video texts.

Course concepts are subject terms in the course. Formally, a course concept c can be defined as a k-gram in $\mathcal{D}$.

External Knowledge is semi-structured data extracted from Wikipedia, denoted as $W = \{w_i\}, i = 1, \cdots, n$. Each wiki-concept w_i in W contains textual content T_i and categories $G_0 = \{g_i\}$. Categories and wiki-concepts have superior-subordinate relationships, we denote g_j is subcategory of g_k as $g_j < g_k$, and $w_j < g_k$ means wiki-concept w_j belongs to category g_k. Since we treat wiki-concepts and categories in the same way in some situation, g_i can also be used to denote wiki-concepts in the rest of this paper. All $g_j < g_i \in G_0$ constitute $G_{-1} = \{g_j | g_j < g_i \; for \; all \; g_i \in G_0\}$, which means all direct descendent categories(or wiki-concepts) of G_0. By that analogy, G_1 is all direct ancestors of G_0, G_2 is all direct ancestors of G_1...

Course concept extraction is formally defined as follows. Given the course corpus $\mathcal{D}$, the objective is to extract candidate course concepts from $\mathcal{D}$, denoted as $\mathcal{T} = \{t_1, \ldots, t_M\}$, and output the concept score s_i for each candidate $t_i \in \mathcal{T}$. s_i indicates the likelihood of t_i to be a course concept in $\mathcal{D}$.

Our challenge is to design a concept score which represents the informativeness of a phrase in a certain area, this will be introduced in section Method. Our model consists of 2 steps: (1) candidate extraction, (2) concepts ranking.

Candidate extraction extracts candidate course concepts from the corpus with heuristic rules. We adopt a POS(Part-of-Speech) pattern: $((Adj|Noun)*(NounPrep)?(Adj|Noun)* |(Adj|Noun)+)Noun$, introduced by [7], to extract all k-gram noun phrases as our candidates. After this step, our manual check shows that more than 98% course concepts are in candidates. Next we should rank candidates to make course concepts top-listed.

Concepts ranking is the most important set of our method, involves ranking the extracted candidates based on their statistics score and knowledge informativeness. Statistics score is a combination of different statistic indicators. These indicators show the combination of each single words in a phrase. Knowledge informativeness is a score calculated by external knowledge W. We use both explicit and implicit representation of w_i to measure the informativeness of a concept. We construct a weighted undirected graph *concept graph* (CG) based on above scores, where each vertex represents a candidate course concept and edges are weighted by the scores. Then we rank the vertexes in the graph via an iterative graph-based propagation algorithm, namely *concept score propagation* (CSP).

IV. Method

We use **CCPEI** (Course Concept Propagation via Explicit/Implicit representation) to represent our method. At first, we introduce statistical and knowledge indicators to help us construct a concept graph.

A. Statistics score

If the constituents of a multi-word candidate phrase form a collocation rather than co-occurring by chance, it is more likely to be considered as a phrase [8]. Referring to [13], We calculate the statistics score Ss for each bi-gram candidate concept d_1, d_2 via the function:

$$Ss(d_1, d_2) = \frac{2 \times freq(d_1, d_2)}{freq(d_1) + freq(d_2)}$$

, where $freq(d)$ represents the frequency of word d in course corpus D. For the N-grams $t = \{d_1, \cdots, d_N\}$, where $N > 2$, the function Ss is defined as

$$Ss(t) = max(\{Ss(f_i, b_i)\}_{i=1,\cdots,N-1}) \tag{1}$$

where $f_i = \{d_1, \cdots, d_i\}$ and $b_i = \{d_{i+1}, \cdots, d_N\}$.

B. Knowledge informativeness

Knowledge informativeness consist of 2 parts: implicit representation and explicit representation.

Implicit representation The informativeness feature of a candidate concept is dependent on its semantic meaning. Due to the problems of low frequency and short context, statistics scores do not provide sufficient information for the semantic meaning of each candidate concept. Thus, we propose to perform semantic representation learning via incorporating information from external knowledge bases.

Distributed representation of words, namely word embeddings [10], have achieved remarkable success in semantic representation of words. Word embeddings represents each word as a low-dimensional, real-valued vector and the semantic similarity between two words can be reflected by the cosine distance of their vectors. We could apply a simple alternative method to obtain semantic representations for candidate course concepts. First, we train word embeddings on Wikipedia corpus. Then, for each candidate concept that consists of L words $d_1, d_2, \cdots d_L$, its vector vp is $vp = v_1 + v_2 + \cdots + v_L$, where v_j is the word vector of d_j.

According to [10], if the vector of a phrase is the summation of word vectors of its individual words, the phrase vector will still encode its semantic information in some way. In the case of course concepts, this property is substantially enhanced because many scientific concepts comprise a meaning that is a simple composition of the meanings of its individual words. For example, $vp(data\ \ mining) = v(data) + v(mining)$ and $vp(machine\ learning) = v(machine) + v(learning)$, and these two phrase vectors have a strong cosine similarity. However, for those phrases having more complex meaning, this simple method may fail to precisely model their semantics. For example, *Occam's Razor* is a domain concept for machine learning that has a slight semantic relation with the word *razor*. But it will have a high cosine similarity with words such as *scissor* and *ruler* if we simply add $v(Occam's)$ and $v(Razor)$ to form its phrase vector.

To address the problem described above, we propose a method that identifies phrases before training and replace them as unique tokens in the training data. Both the course corpus and Wikipedia provide plenteous information for phrase identification. Candidate concepts with high Statistics score is considered to be phrases. Wikipedia articles also contain a wealth of human-created hyperlinks, which can be regarded as human-labeled phrases. We formally define our method with the following notation:

$T = \{t_i, Ss(t_i)\}$: candidate course concept set.

$H = \{h_i, freq(h_i)\}$: hyperlink set of Wikipedia, h_i is the text of the hyperlink and $freq(h_i)$ denotes its occurrence frequency.

Then, we create the phrase set P, which is a subset of T and defined as follows:

$$\begin{aligned} U &= \{t | Ss(t) > \phi\} \\ V &= \{t | t \in H \ and \ freq(t) > Min\} \\ P &= T \cap (U \cup V) \end{aligned} \tag{2}$$

where ϕ and Min are two pre-set thresholds for selecting phrases which is set as 0.6 and 10 respectively in our experiments. We replace the phrases in P as unique tokens in the encyclopedia corpus to obtain our training corpus and train word embeddings on it. After training, the concepts in P will have their own vector representations. For the candidates that are not in P, we obtain its vector via the summation of its word vectors as described above.

After above processing, we get implicit representation of a candidate concept as a vector. Then we use cosine similarity as the implicit similarity $Sim_I(t_i, t_j)$ of two concepts.

Explicit representation Explicit representation mines the categories information in external knowledge. The superior-subordinate relationship of categories construct a directed acyclic graph. For a certain wiki-concept w_i, its categories are some vertexes in the graph. If we consider the ancestors and descendents of these vertexes, we can get a subgraph of the whole graph generated from all external knowledge W. This subgraph is a representation of wiki-concept w_i. The explicit representation Er of w_i is defined as following:

$$Er = \{G_k | k \in Z\}$$

, In our experiments, value of k is limited in $\{-2, -1, 0, 1, 2\}$. Fig. 2 shows an example of explicit representation.

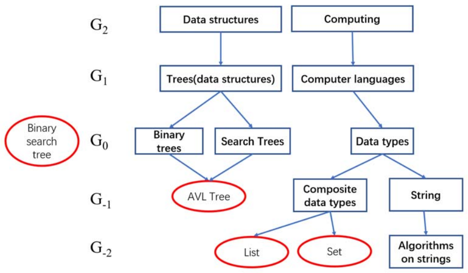


Fig. 2. Explicit representation of concept Binary Search Tree. concepts are in red circles, categories are in blue rectangles.

For two wiki-concepts w_i and w_j, we can use Er_i and Er_j to calculate a explicit similarity Sim_E between them. For all g_k which $g_k \in Er_i$ and $g_k \in Er_j$, we have following formula:

$$Sim_E(w_i, w_j) = \sum_k wei(g_k, Er_i) * wei(g_k, Er_j)$$

, where $wei(g_k, Er_i)$ means:

$$wei(g_k, Er_i) = \begin{cases} 1.0 & g_k \in G_{-2} \subset Er_i \\ 1.25 & g_k \in G_{-1} \subset Er_i \\ 9.75 & g_k \in G_0 \subset Er_i \\ 4.75 & g_k \in G_1 \subset Er_i \\ 2.5 & g_k \in G_2 \subset Er_i \end{cases} \tag{3}$$

Values in $wei(g_k, Er_i)$ are chosen by grid searching(fix weight of G_{-2} at 1.0, search others in [-5,15] with step length 0.25). All Sim_E in once extraction task will be normalized, the largest one is scaled to 1.0, others are scaled with the same scaling ratio.

Obviously, candidate concepts are usually not in the external knowledge W. If t_i is not in W, we will check postfixes of t_i, pick up the longest postfix that can find in W(be called as w_i) to calculate similarity between t_i and a wiki-concept w_j: $Sim_E(t_i, w_j) = \beta Sim(w_i, w_j)$. In this occasion, $Sim_E(w_i, w_j)$ should be multiplied by a penalty ratio β, which is the word length ratio of w_i and t_i. For instance, candidate concept *graph convolutional network* can not be found in W, but its postfix *convolutional network* is in W. Then $Sim_E(graph\ convolutioanal\ network, t_j) = \frac{2}{3} Sim_E(convolutioanal\ network, t_j)$.

After getting two similarities Sim_I and Sim_E, we ensemble them as the similarity between 2 candidate concepts: $Sim(t_i, t_j) = \alpha Sim_I(t_i, t_j) + (1-\alpha) Sim_E(t_i, t_j)$. α is a hyper-parameter. We set it on 0.4 in our experiments.

C. Concept graph construction

In common graph-based methods such as TextRank [9], the graph is constructed based on co-occurrence relation between words. However, local context features such as co-occurrence can hardly be utilized in our task because of the short context problem. In our model, we base on the implicit/explicit representations of candidate concepts to construct the course concept graph.

Concept Graph.A concept graph is a weighted undirected graph denoted as $G = (V, E)$. Each candidate course concept t_i is a vertex in CCG and an edge $(t_i, t_j) \in E$ if $Sim(t_i, t_j) > \theta$, where θ is a pre-set threshold, and $Sim(v_i, v_j)$ is the cosine distance between the vectors of t_i and t_j. The weight of an edge $e(t_i, t_j) = Sim(v_i, v_j)$.

D. concept score propagation

Concept score propagation is a iterative algorithm on concept graph. Every vertexes in the graph will get a score after the algorithm. We denote the score of the concept vertex t after the k-th iteration as $score^k(t)$ and the initial score of concept t as $score^0(t)$. In our model, $score^0(t_i)$ is obtained using a seed set. The seed set, which is denoted as S, contains a list of human-labeled course concepts. The concepts in the seed set are usually a few core concepts of this course which can be easily obtained from chapter list of a course. We simply set $score^0(t_i) = 1$ for $t_i \in S$ and $score^0(t_i) = 0$, otherwise. These seeds are hot spots that connect with multiple other course concepts in the graph. Thus, we set them as start points for propagation. After several iterations of propagations, their scores will be propagated to other course concepts in the graph.

Propagation Process. In each iteration, the score of a vertex t_i is calculated as

$$\begin{aligned} score^{k+1}(t_i) &= \frac{1}{Z} \cdot \frac{1}{|I(t_i)|} \times \\ \sum_{t_j \in I(t_i)} & Ss(t_j) \cdot Sim(t_j, t_i) \cdot score^k(t_j) \end{aligned} \tag{4}$$

where $I(t_i)$ is the set of vertexes that is adjacent to t_i, Z is the normalization factor.

When the average score of the chosen seed set has a significant decline(descend to 0.7 or lower), the iteration will be stopped at the last round. For a course concept, it will have many course concept neighbors that provide score for it. After several iterations, these course concepts will all have a relatively high score.

The overlapping problem in propagation. If two candidate concepts t_i and t_j contain one or more identical grams, we say that t_i and t_j are *overlapping*. For example, *merge sort* and *bubble sort* are overlapping because they both have the gram *sort*. Overlapping frequently occurs among course concepts because scientific concepts often have background words such as *function*, *algorithm* and *culture*. We observe that overlapping between concepts may have a negative effect on our propagation process.

For example, the concept *same algorithm* is not a course concept but contains the word *algorithm*, which enables it to have a high cosine similarity with course concepts that include the word *algorithm*, such as the *bfs algorithm* and the *kmp algorithm*. Thus, the score of the concept *same algorithm* will be blindly increased by votes from these course concept neighbors. Essentially this problem arise because the cosine similarity of implicit representation between overlapping concepts is unable to reflect their real semantic relation. Thus, we add a patch to our propagation algorithm to solve this problem.

Fig. 1 shows the solution of overlapping problem. When a candidate concept has overlapping adjacent concepts, all scores come from them will be averaged, and only contribute once for the score. We call this solution **overlapping merge**.

V. Experiments

A. Dataset

XuetangX[1] is the most popular MOOC website in China, which offers 1,300 courses for 9 million users by November 2017. Based on XuetangX data, we select courses with different domains and languages to form our three

[1] www.xuetangx.com

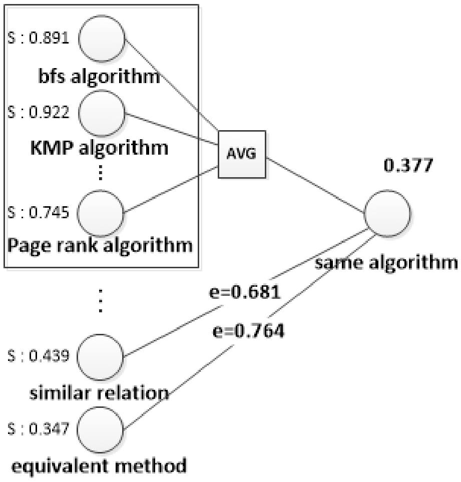


Fig. 3. Solution of overlapping problem: overlapping merge.

evaluation datasets: CSZH(computer science in Chinese), EcoZH(economics in Chinese) and CSEN(computer science in English). For the Chinese datasets—CSZH and EcoZH, we employ Ansj, which is a state-of-the-art Chinese word segmentation system to perform word segmentation and POS tagging.[2] For the English dataset—CSEN, we select the POS tagger implemented by the Stanford NLP group.[3] The statistic information for our evaluation datasets are summarized in Table I.

TABLE I
DATASET STATISTICS

Dataset	CSZH	EcoZH	CSEN
domain	CS	Economics	CS
language	Chinese	English	Chinese
courses	18	6	5
videos	1,620	468	571
text size	2.9M	1.2M	1.3M

Human annotations have been performed on the datasets to label the course concepts. For each dataset, Human annotations have been performed by four graduate students majoring in the corresponding domain. Each annotator is presented with a candidate concept list, and is asked to give numeric judgments in the interval [0,100] for each candidate. A candidate is labeled as a course concept if its average score is greater than 75. On a certain concept, if all annotators are in agreement, we will adopt it as a course concept.

B. Comparison methods and Evaluation

We compare our model with the following methods:

TF-IDF: TF-IDF [1] is a widely used statistical-based method for key-phrase extraction. In our study, TF is computed based on candidate frequency in the given video, and IDF is calculated by the number of videos in which the candidate appears.

TextRank: TextRank [9] is one of the most well-known graph-based approaches to keyphrase extraction. It represents a document as a graph, where vertexes represent words and edges are connected based on word co-occurrence relations in a fixed-length window. The PageRank [11] algorithm is applied for scoring vertexes, and the score of a phrase is the average score of its tokens. We use TextRank4ZH, which is a python implementation of TextRank, for our experiments. [4]

CCP: This is our previous work [12], also use graph propagation method to solve the concept extraction problem.

Evaluation Metric. We choose Mean Average Precision(MAP) as our evaluation metric. MAP is the standard single-number measure for comparing search algorithms, which considers the position of ranking items. Since results of all comparison methods are presented as a ranking format, this metric can reveal the ranking performance.

In real application, concepts extracted by algorithms need manual filtering to guarantee the effect showed to users. In this situation, the most important indicator of performance is recall of concepts in the top. Because annotators can help to delete inaccurate concepts in a hundred scale list, but hard to pick up concepts from an interminable list. Thus we use recall of top 10%(R@10%) as the second evaluation metric.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

The comparison among different methods is shown in Table II. The proposed CCPEI method outperforms the comparison methods in Chinese datasets, which is the main language that we tend to apply our method.

TABLE II
PERFORMANCE OF DIFFERENT METHODS ON THE THREE DATASETS(%)

corpus	method	MAP	R@10%
CSZH	TF-IDF	11.7	26.8
	TextRank	15.1	31.9
	CCP	41.3	58.1
	CCPEI	**47.5**	**61.4**
EcoZH	TF-IDF	18.8	27.9
	TextRank	17.0	30.6
	CCP	43.2	64.2
	CCPEI	**48.6**	**66.8**
CSEN	TF-IDF	12.4	27.2
	TextRank	14.9	37.7
	CCP	**42.5**	**60.2**
	CCPEI	41.7	53.7

A. Performance on English data.

However, CCPEI does not perform best on English dataset. After we deleted the explicit part of CCPEI, it get a result with 0.430 MAP. This shows that explicit representation has negative effect on English.

To find the reason, we make some manual check on the differences between Chinese and English Wikipedia categories.

[2] https://github.com/NLPchina/ansjseg

[3] http://nlp.stanford.edu/software/tagger.shtml

[4] https://github.com/someus/TextRank4ZH

We find that although English concepts in Wikipedia has more average categories than Chinese concepts, but some categories do not help to concept extraction. For example, the concept “AVL Tree” has 4 categories: “1962 in computer science”, “Binary trees”, “Soviet inventions” and “Search trees”. The “Soviet inventions” category is not related to computer science, then makes some concepts in other areas(such as “Laser”) contribute scores to it. But in Chinese Wikipedia, “AVL Tree” has only one category — “Tree structure”. Thus it will not receive score from irrelevant areas. After we manually deleted some irrelevant categories in English Wikipedia data, MAP is raised to 43.7% and R@10% is raised to 59.3%. It reveals that if we want to make the performance better on English data, we need to introduce a category filter to provide above situation.

B. *Study of the overlapping problem.*

We study whether the overlapping merge benefits the course concept extraction. We manually pick up 135 concepts containing the word “sort” from candidate concepts of CSZH, and 15 of them are course concepts. In the ranking result of our method without overlapping merge, mean rank of the 15 course concepts is 252, and mean rank of the other 120 negative candidates is 2619. With regard to overlapping merge, the mean ranks are 243 and 6174, respectively. It indicates that overlapping merge alleviates the overlapping problem without losing predictive ability of real course concepts.

C. *Contribution of Different Parts.*

CCPEI consists of some low-coupling parts. In experiments, we can evaluate different parts by remove one of them. Fig. 4 shows the MAP on CSZH when remove one part of CCPEI.In the figure, origin means remove nothing, statistic means statistic score, and so on. It is obvious that implicit representation contribute most in CCPEI. And other parts contribute the final performance respectively.

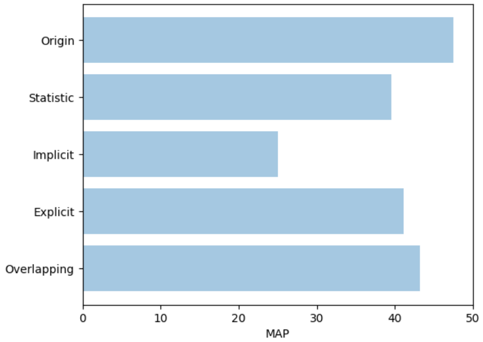


Fig. 4. Contribution of different parts in CCPEI(MAP)

VII. Conclusion and Future Work

In this paper, we study the problem of course concept extraction in MOOCs. With the help of external knowledge in Wikipedia, we define the problem and propose a explicit/implicit representation based propagation method to extract course concepts. Experiment on three different datasets validate the effectiveness of the proposed method.

In our analysis, we find that qualities of categories determine the effect of explicit representation. Some techniques should be designed to automatically filter categories to get better explicit representation. Meanwhile, since categories are not the only structured data in Wikipedia, more structured data such as infoboxes can be used to generate better concept representation.

References

[1] Salton, Gerard, and Christopher Buckley. ”Term-weighting approaches in automatic text retrieval.” Information processing and management 24.5 (1988): 513-523.

[2] Frantzi, Katerina, Sophia Ananiadou, and Hideki Mima. ”Automatic recognition of multi-word terms:. the c-value/nc-value method.” International journal on digital libraries 3.2 (2000): 115-130.

[3] Matsui, K., et al. ”A lipid-hydrolysing activity involved in hexenal formation.” (2000): 857-860.

[4] Chang, Jing-Shin. ”Domain specific word extraction from hierarchical Web documents: A first step toward building lexicon trees from Web corpora.” Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing. 2005.

[5] Lehmann, Jens, et al. ”DBpediaa large-scale, multilingual knowledge base extracted from Wikipedia.” Semantic Web 6.2 (2015): 167-195.

[6] Liu, Zhiyuan, et al. ”Automatic keyphrase extraction via topic decomposition.” Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2010.

[7] Justeson, John S., and Slava M. Katz. ”Technical terminology: some linguistic properties and an algorithm for identification in text.” Natural language engineering 1.1 (1995): 9-27.

[8] Korkontzelos, Ioannis, Ioannis P. Klapaftis, and Suresh Manandhar. ”Reviewing and evaluating automatic term recognition techniques.” Advances in natural language processing. Springer, Berlin, Heidelberg, 2008. 248-259.

[9] Mihalcea, Rada, and Paul Tarau. ”Textrank: Bringing order into text.” Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

[10] Mikolov, Tomas, et al. ”Distributed representations of words and phrases and their compositionality.” Advances in neural information processing systems. 2013.

[11] Page, Lawrence, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.

[12] Pan, Liangming, et al. ”Course Concept Extraction in MOOCs via Embedding-Based Graph Propagation.” Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Vol. 1. 2017.

[13] Church, Kenneth Ward, and Patrick Hanks. ”Word association norms, mutual information, and lexicography.” Computational linguistics 16.1 (1990): 22-29.

[14] Turney, Peter D. ”Learning algorithms for keyphrase extraction.” Information retrieval 2.4 (2000): 303-336.

[15] You, Wei, Dominique Fontaine, and Jean-Paul Barths. ”An automatic keyphrase extraction system for scientific documents.” Knowledge and information systems 34.3 (2013): 691-724.

[16] Frank, Eibe, et al. ”Domain-specific keyphrase extraction.” 16th International Joint Conference on Artificial Intelligence (IJCAI 99). Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. APA

[17] Witten, Ian H., et al. ”KEA: Practical automatic keyphrase extraction.” Proceedings of the fourth ACM conference on Digital libraries. ACM, 1999.

[18] Medelyan, Olena, and Ian H. Witten. ”Thesaurus based automatic keyphrase indexing.” Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. ACM, 2006.

[19] Yih, Wen-tau, Joshua Goodman, and Vitor R. Carvalho. ”Finding advertising keywords on web pages.” Proceedings of the 15th international conference on World Wide Web. ACM, 2006.

[20] Kim, Su Nam, and Min-Yen Kan. ”Re-examining automatic keyphrase extraction approaches in scientific articles.” Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications. Association for Computational Linguistics, 2009.

[21] Lopez, Patrice, and Laurent Romary. ”HUMB: Automatic key term extraction from scientific articles in GROBID.” Proceedings of the 5th international workshop on semantic evaluation. Association for Computational Linguistics, 2010.

[22] Wan, Xiaojun, and Jianguo Xiao. ”Single Document Keyphrase Extraction Using Neighborhood Knowledge.” AAAI. Vol. 8. 2008.

[23] Huang, Chong, et al. ”Keyphrase extraction using semantic networks structure analysis.” Data Mining, 2006. ICDM’06. Sixth International Conference on. IEEE, 2006.

[24] Liu, Jialu, et al. ”Mining quality phrases from massive text corpora.” Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM, 2015.

[25] Gazendam, Luit, Christian Wartena, and Rogier Brussee. ”Thesaurus based term ranking for keyword extraction.” Database and Expert Systems Applications (DEXA), 2010 Workshop on. IEEE, 2010.

[26] Rospocher, Marco, et al. ”Corpus-based terminological evaluation of ontologies.” Applied Ontology 7.4 (2012): 429-448.

[27] Vivaldi, Jorge, and Horacio Rodrguez. ”Finding Domain Terms using Wikipedia.” LREC. 2010.

[28] Berend, Gbor, and Richrd Farkas. ”SZTERGAK: Feature engineering for keyphrase extraction.” Proceedings of the 5th international workshop on semantic evaluation. Association for Computational Linguistics, 2010.