SemKeyphrase: An Unsupervised Approach to Keyphrase Extraction from MOOC Video Lectures

Abdulaziz Albahr
Southern Illinois University
Carbondale, IL, USA
a.albahr@cs.siu.edu

Dunren Che
Southern Illinois University
Carbondale, IL, USA
dche@cs.siu.edu

Marwan Albahar
Umm Al-Qura University
Makkah, Saudi Arabia
marwanalialbahar@gmail.com

ABSTRACT

The Massive Open Online Courses (MOOCs) have emerged as a great resource for learners. Numerous challenges remain to be addressed in order to make MOOCs more useful and convenient for learners. One such challenge is how to automatically extract a set of keyphrases from MOOC video lectures that can help students quickly identify a suitable knowledge without spending too much time and expedite their learning process. In this paper, we propose *SemKeyphrase*, an unsupervised cluster-based approach for keyphrase extraction from MOOC video lectures. *SemKeyphrase* incorporates a new ranking algorithm, called *PhaseRank*, that involves two phases on ranking candidate keyphrases. Experiment results on a real-world dataset of MOOC video lectures show that our proposed approach outperforms the state-of-the-art methods by 16% or more in terms of F_1 score.

KEYWORDS

MOOCs, Automatic keyphrase extraction, Unsupervised, Cluster-based

ACM Reference Format:
Abdulaziz Albahr, Dunren Che, and Marwan Albahar. 2019. SemKeyphrase: An Unsupervised Approach to Keyphrase Extraction from MOOC Video Lectures. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19), October 14–17, 2019, Thessaloniki, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3350546.3352535

1 INTRODUCTION

In MOOC platform (e.g.,Coursera), course design typically follows a linear structure, in which video lectures are presented sequentially on weekly bases and are associated with quizzes at the end of a week [3] [5] [9]. Additionally, unlike other structured educational resources such as books, the content of video lectures can only be navigated in linear way [1] [29]. Consequently, learners, new to a subject or interested in a specific concept, may find it challenging to know the main knowledge addressed in MOOC video lectures and to navigate among and find the right content matching their learning goals. Automatic provision of keyphrases that precisely capture the main knowledge of MOOC video lectures can greatly help mitigate the challenge. With the keyphrases, we can generate a table of content of video lectures in a course that help learners efficiently navigate to desired parts of MOOC video lectures. With the help of such a table of content, learners can quickly navigate to the relevant places to review when preparing an exam or test, which is the most common type of non-linear navigation desired and performed by learners of MOOCs according to [9]. However, MOOCs come without keyphrases, and identifying keyphrases manually for MOOCs is tedious, time-consuming and costly. Thus, automatic keyphrase extraction (AKE), which is a task of automatically identifying important phrases from the content of a document [24][8], is essential for MOOC video lectures.

In this paper, We present a novel unsupervised cluster-based approach for keyphrases extraction from MOOC video lectures, called *SemKeyphrase*, that incorporates a new ranking algorithm, called *PhaseRank*, which plays a key role in our approach. The PhaseRank consists of two phases of process: ranking of candidate keyphrase clusters and ranking of top candidate keyphrases. Our experiment on a real-world dataset of MOOC video lectures show that our SemKeyphrase approach yields an improvement of 16% and 12% in term of F_1 and *MRR* scores respectively over the commonly used baseline extraction and state-of-the-art algorithm for keyphrase extraction.

The rest of the paper is organized as follow: Section 2 reviews the state-of-the-art methods of keyphrase extraction. Section 3 provides an overview of our proposed approach, *SemKeyphrase*. Our Proposed ranking algorithm is explained in section 4. Section 5 shows our experimental result, and Section 6 concludes the paper.

2 RELATED WORK

This paper is related to automatic keyphrase extraction approaches. Generally, automatic keyphrase extraction approaches can be divided into two groups: supervised and unsupervised approaches [10] [11]. In the supervised approaches, keyphrase extraction is defined as a binary classification problem, where a candidate keyphrase is classified as either a keyphrase or non-keyphrase [24][12][28][21][15][27][4] based on various features such as TF-IDF and length of phrase, leveraging different machine learning algorithms such as Naive Bayes [11].

In unsupervised approaches, keyphrase extraction is modeled as a ranking problem; unsupervised approaches can be further be divided into three subgroups: statistical, graph-based, and cluster-based approaches [10][11]. Statistical approaches represent a document as a bag of words and use statistical measures to associate each word with a score that is later used to distinguish keyphrases from non-keyphrases. The most common statistical metric used in this group is the TF-IDF metric.

WI '19, October 14–17, 2019, Thessaloniki, Greece

ACM ISBN 978-1-4503-6934-3/19/10…$15.00
https://doi.org/10.1145/3350546.3352535

Graph-based approaches model a given document as a graph and rank its vertices (words) based on their significance using graph-based ranking algorithms [11]. TextRank [18] is a representative graph-based approach. Other approaches based on extending TextRank have been proposed such as SingleRank [25], ExpandRank [25], and Topical PageRank (TPR) [13]. Recently, Florescu and Caragea [6] proposed PositionRank that integrates the frequency and position of a word's occurrence into a biased PageRank algorithm to extract keyphrases from research papers. Several graph-based approaches [17] exploit external knowledge bases and word embedding vectors to improve the performance of keyphrase extraction [17] [26] [22].

Cluster-based approaches first group similar candidate phrases into clusters using different semantic relatedness metrics and then select one representative candidate phrase from each cluster to facilitate extracting keyphrases [10][11]. Liu et al. [14] proposed KeyCluster that leverages clustering techniques. Bougouin et al [2] proposed TopicRank that combines graph and clustering techniques.

3 PROBLEM FORMULATION AND OVERALL APPROACH

In this section, we first define the problem of keyphrase extraction from MOOC video lectures and then present our unsupervised cluster-based approach at a high level. We formally define the problem of keyphrase extraction from MOOC video lectures as follows: Let $D = \{V_{11}, ..., V_{1N}, V_{21}, ..., V_{MN}\}$ be a collection of MOOC video lectures representing M MOOCs in the same domain such as computer science, where V_{ij} represents j^{th} video lecture in a MOOC course i. N denotes the total number of video lectures of the course and M denotes the total number of MOOCs. Each V_{ij} consists of video lecture title t_{ij} and video lecture text d_{ij}. Our overall approach, SemKeyphrase, takes D as input and for each $V_{ij} \in D$ generates a set of salient keyphrases that can represent V_{ij}.

Our overall approach takes a "pipeline" structure, consisting of four main steps/components: (1) candidate keyphrases extraction, (2) semantic relatedness computation, (3) candidate Keyphrases clustering, and (4) candidate keyphrase ranking.

The candidate keyphrase extraction step performs text preprocessing on input lecture texts (sentence boundary detection, tokenization, part-of-speech (POS) tagging), and then selects a list of initial candidate keyphrases. The semantic relatedness computation step computes a semantic relatedness score between the selected candidate keyphrases using a new semantic relatedness metric. The candidate keyphrases clustering step groups the selected candidate keyphrases into clusters based on their computed semantic relatedness scores. In the candidate keyphrase ranking step, a newly-designed ranking algorithm, PhaseRank, consisting of two phases is used to rank the selected candidate keyphrases. The first three steps our approach are expounded in the subsequent subsections in turn, while the details of candidate keyphrase ranking step is presented in a separate subsequent section.

3.1 Candidate Keyphrases Extraction

Let V_{ij} be a target MOOC video lecture from which keyphrases are to be extracted. First, the input text is tokenized and annotated with part-of-speech (POS) tags using the Stanford NLP toolkit [16]. Then, we adopt a similar method as in [18][25] to extract all n-gram noun phrases as our initial candidate keyphrases matching the regular expression $(JJ|RB) * (NN)+$ as most manually identified keyphrases are noun phrase [12]. In the regular expression, JJ, RB, NN indicate adjective, adverb, and noun, respectively. The result of this step is a list of initial candidate keyphrases formally denoted by $CP_{ij} = \{p_1, p_2, .., p_s\}$.

3.2 Semantic Relatedness Computation

This step computes a semantic relatedness score for each candidate keyphrase in regard to all other candidate keyphrases. Similar to [26][22], our method first use the skip-gram model proposed by Mikolov et al. [19][20] that utilizes Wikipedia data corpus as context (i.e. providing an example for each candidate keyphrase) to learn the word embedding vector for each candidate keyphrase, and then compute the semantic relatedness between each pair of candidate keyphrases (p_k, p_j) using a novel semantic relatedness metric.

Our a new metric defines the semantic relatedness score between two candidate keyphrases, denoted as $SR(p_k, p_j)$, as the cosine similarity between their corresponding word embedding vectors plus the co-occurrence counts of the two candidate keyphrases within a window of w successive words in the MOOC video lecture multiply by their strength of semantic relation to compensate the low frequency of their co-occurrence relation.

3.3 Candidate Keyphrases Clustering

Inspired by [14], the list of initial candidate keyphrases CP_{ij} are grouped into a set of clusters $C = \{c_1, c_2, .., c_q\}$ based on the semantic relatedness scores computed in section 3.2, where c_i is the i-th cluster in C. Each cluster c_i is composed of semantically related candidate keyphrases. We adopt Affinity Propagation (AP) [7] since AP is a handy and most widely accepted clustering algorithm [14], but our approach is open to other suitable clustering algorithms.

4 CANDIDATE KEYPHRASE RANKING

We propose a new ranking algorithm, called PhaseRank, to rank the candidate keyphrases which have been clustered at the last step of our overall approach. PhaseRank consists of two ranking phases: (1) ranking the clusters and generating initial ranked top candidate keyphrases; (2) reranking the top candidate keyphrases and producing final output keyphrases. The two phases are described in detail in the following subsections.

4.1 Ranking Clusters

We rank our generated clusters of candidate keyphrases based on their measured importance to the MOOC video lecture from which the candidate keyphrases were extracted. We take it for granted that every MOOC video lecture has a main (or thematic) topic, which comprises a number of subtopics, and each subtopic is embodied by a section (segment) of the lecture text, i.e., a set of adjacent sentences. In this work, we have no need to explicitly characterize/represent the subtopics, but just to acknowledge their existence (explicit or implicit). Naturally, for a given subtopic, some phrases are more informative and important than others. Our cluster ranking method is based on the assumption that a more important cluster

would contain more candidate keyphrases that are important to the subtopics of a MOOC video lecture.

Our cluster ranking method proceeds by, first, computing the importance score of each candidate keyphrase with regard to each subtopic following our above observation and assumption; second, deciding the significance score of each cluster with regard to the MOOC video lecture; third, determining the "semi-final" list of candidate keyphrases, referred to as the initial ranked list of candidate keyphrases.

4.1.1 Candidate Importance Score. A candidate keyphrase is more important to a subtopic when the candidate keyphrase has stronger semantic relations with most of the candidate keyphrases involved in the subtopic. More specifically, a candidate keyphrase is important to a subtopic if its average score of semantic relatedness to all other candidate keyphrases in the subtopic is higher than the average pairwise semantic relatedness score of all candidate keyphrases in the subtopic. Let PS_{imp} be the score representing the importance of a candidate keyphrase to a subtopic. To compute PS_{imp}, our method divides a MOOC video lecture text d_{ij} into H sections (segments), and each section, denoted by $h_i = \{s_1, s_2, ..., s_l\}$, consists of a sequence of l sentences. l is experimentally set. For each section h_i, the section's relevant candidate keyphrases, denoted as $h_i' = \{p_1, ..., p_z\}$, are identified. *Section h_i's average semantic relatedness score* is computed by the following equation:

$$SectionSem(h_i) = \frac{\sum_{i,j=1,i\neq j}^{z} SR(p_i, p_j)}{z \times (z-1)} \quad (1)$$

Keyphrase p_k's average semantic relatedness score with regard to section h_i is computed by the following equation:

$$AveSem(p_k, h_i) = \frac{\sum_{j=1,k\neq j}^{z} SR(p_k, p_j)}{z} \quad (2)$$

We now compute PS_{imp}, the importance score of p_k to the subtopic of section h_i, using the following equation:

$$PS_{imp}(p_k, h_i) = \frac{AveSem(p_k, h_i)}{SectionSem(h_i)} \times wnum(p_k) \quad (3)$$

Where $wnum(p_k)$ stands for the number of words in candidate keyphrase p_k.

For each candidate keyphrase $p_k \in CP_{ij}$, we find all the sections where p_k appears, and denote p_k's *relevant section set* as $Sec(p_k) = \{h_i | 1 \leq i \leq H\}$. The final importance score FPS_{imp} of the candidate keyphrase p_k takes the highest $PS_{imp}(p_k, h_i)$ score over all p_k's relevant sections as follows.

$$FPS_{imp}(p_k) = \max_{h_i \in Sec(p_k)} \{PS_{imp}(p_k, h_i)\} \quad (4)$$

4.1.2 Cluster Significance Score. Our method ranks candidate keyphrase clusters per cluster significance scores. We introduce cluster significance score $Cluster_{sig}$ to represent the significance of a cluster (of candidate keyphrases) to the MOOC video lecture. Intuitively, the significance of cluster would increase if the cluster contains more candidate keyphrases with high FPS_{imp} scores. Accordingly we define cluster significance score by the following equation:

$$Cluster_{sig}(c_i) = \sum_{p_k \in c_i} FPS_{imp}(p_k) \times Overlap(p_k, c_i) \quad (5)$$

Where $Overlap(p_k, c_i)$ imposes a penalty when a candidate keyphrase p_k overlaps with another candidate keyphrase p_j in the same cluster c_i. It is set to 1 if there is no overlap, otherwise to $\alpha \in [0, 1]$. Here, α is to be experimentally determined.

4.1.3 Initial Ranked List of Candidate Keyphrases. We take the top-T clusters and from them we generate a initial ranked list of top candidate keyphrases denoted as $CandidateRank_{initial} = \{p_1, p_2, ..., p_u\} \subset CP_{ij}$. We select candidate keyphrases from the top-T clusters to form $CandidateRank_{initial}$ using a new selection strategy. The proposed selection strategy chooses a centroid of each cluster in the top-T clusters list as well as other candidate keyphrases that have FPS_{imp} score above the average FPS_{imp} score of all candidate keyphrases in the cluster.

4.2 Reranking Candidate Keyphrases

In the second phase, the candidate keyphrases generated by the last phase is reranked using more sophisticated ranking criteria in order to generate really meaningful keyphrases as final output. Therefore, we propose a new ranking metric called $P_{significance}$ that incorporates three aspects of quality of a candidate keyphrase for being a good keyphrase of a given MOOC video lecture. To our knowledge, such ranking formula has not been investigated elsewhere. After all the selected candidate keyphrases are reranked according to their $P_{significance}$ scores, the top-k keyphrases in the newly ranked list of candidate keyphrases are then output as the final top-k keyphrases of the MOOC video lecture (k being the desired number of keyphrases of a MOOC video lecture).

The three aspects of quality of a candidate keyphrase incorporated in the $P_{significance}$ score include: (1) title semantic relatedness score CT_{Sem} that measures the semantic relatedness of a candidate to the MOOC lecture's title phrases, (2) other candidates relatedness score CC_{Sem} that measures the semantic relatedness of the candidate to other candidate keyphrases in the generated initial ranked list, and (3) phraseness score Phr that measures the candidate's suitability for being a "valid" keyphrase.

For each candidate keyphrase, we compute CT_{Sem},CC_{Sem}, and $Phr(p_k)$ scores and then combine the three scores into one equation to compute $P_{significance}$ as follow:

$$P_{significance}(p_k) = CT_{Sem}(p_k) \times CC_{Sem}(p_k) \times Phr(p_k) \quad (6)$$

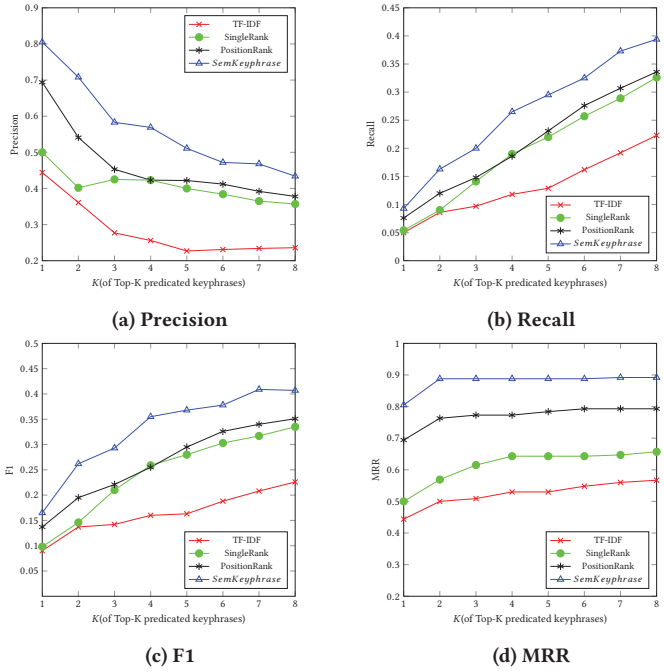
After the candidate keyphrases are reranked per $P_{significance}$ scores, we retain the top-k candidate keyphrases in the ranked list for final output.

5 EXPERIMENTS AND RESULTS

In this section, we first describe the construction of MOOCs dataset as well as evaluation metrics we are to use. Then, we study the performance of SemKeyphrase method comparing with commonly used baselines and the state-of-the-art method.

5.1 Dataset and Evaluation Metrics

For this experimental study, we build a real-word dataset of MOOC video lectures collected from a famous MOOC platform, Coursera, as currently there is no publicly available dataset that fits the need of evaluating our SemKeyphrase method of extracting keyphrases for individual MOOC video lectures rather than the whole MOOC



(a) Precision (b) Recall

(c) F1 (d) MRR

Figure 1: Performance Comparison of SemKeyphrase vs. baselines and the state-of-the-art method.

Table 1: Performance data (SemKeyphrase vs. KeyCluster)

Methods	Precision	Recall	F1
KeyCluster	0.1551	0.6536	0.2427
SemKeyphrase with k= 8	0.434	0.394	**0.407**

courses [22]. Our dataset consists of 30 MOOC video lectures in the discipline of computer sciences extracted from Coursera. Each MOOC video lecture consists of a transcript (i.e. lecture text) and a title, and all these MOOC video lectures are in English language. A gold-standard of manually labeled keyphrases for evaluation was made as follows. For each MOOC video lecture, a list of all candidate keyphrases generated using our method described in section 3.1 was presented to two human experts in the discipline to judge whether the candidate keyphrases are suitable keyphrases for the MOOC video lecture according to the transcript and the title of the MOOC video lecture. A candidate keyphrase is finally labeled as keyphrase of a MOOC video lecture if both experts agreed on.

Like other related works [25][13][18][11], we adopt the three common evaluation metrics used for keyphrase extraction: *Precision*, *Recall*, and *F*1. In addition, we use *mean reciprocal rank* to evaluate the order of our finally generated top-*k* ranked list of predicated keyphrases (*k* ranges from 1 to 8 in our setting).

5.2 Performance Comparison

We compare our SemKeyphrase with the most three commonly used baselines and the state-of-the-art method relevant to our work, i.e., TF-IDF [11], SingleRank [25], KeyCluster [14], and PositionRank [6]. Our selection of these baselines is based on prior related works [13][25][14], especially the very popular survey on keyphrase extraction [11]. For SingleRank and PositionRank, we use the implementation provided by [6]. We use the skip-gram model with default parameter from genism python library[2] to train word embedding and the Affinity Propagation(AP) algorithm from scikit-learn python library [23] for clustering the candidate keyphrases. A Wikipedia dump[3] dataset is used for training word embedding and computing the *phraseness* scores.

To demonstrate the effectiveness of our approach, we designed two experiments. In the first experiment, we compare SemKeyphrase with KeyCluster [10], a cluster-based approach that does not use any ranking algorithm to generate keyphrases. In the second experiment, we compare SemKeyphrase with two other common baselines and the state-of-the-art method, i.e., TF-IDF, SingleRank, and PositionRank.

We first compare our cluster-based approach, SemKeyphrase, with cluster-based approach, KeyCluster [10]. Table 1 shows the result of the comparison of SemKeyphrase with KeyCluster in term of *Precision*, *Recall*, and *F*1. As we can see in the table, SemKeyphrase significantly outperforms KeyCluster on *F*1 and *precision* scores. For example, SemKeyphrase achieves an *F*1 score of 0.407 compared to 0.242 achieved by KeyCluster with an improvement of 68%. The improvement is mainly due to the effect of the ranking algorithm in our approach that has no counterpart in KeyCluster.

We next compare SemKeyphrase with two other baselines and the state-of-the-art method, i.e., TF-IDF, SingleRank, and PositionRank. Fig.1 shows the result of the comparison of SemKeyphrase with TF-IDF, SingleRank, and PositionRank in term of *Precision*, *Recall*, *F*1, and *MRR* for top-*k* predicated keyphrases, with *k* ranging from 1 to 8. From Fig. 1, we can see that SemKeyphrase outperforms all other methods on both *F*1 and *MRR* scores. For example, SemKeyphrase shows an improvement of 16% and 12% in *F*1 and *MRR* scores respectively over PositionRank, which is the best performing competitor.

6 CONCLUSION

In this paper, we have studied the problem of keyphrase extraction from MOOC video lectures. We proposed a novel unsupervised cluster-based approach, called SemKeyphrase, for extracting keyphrases from MOOC video lectures. Our proposed method, SemKeyphrase, incorporates a new ranking algorithm, called PhaseRank, that is particularly suitable for ranking the candidate keyphrases in MOOCs. Our experiments on a real-word dataset of the MOOC video lectures show that our approach far outperforms the three common baselines and the state-of-the-art methods by 16% in term of *F*1 and 12% in term of *MRR*, which sufficiently demonstrate the efficacy and the out-performance of our proposed novel approach.

REFERENCES

[1] Akshay Agrawal, Jagadish Venkatraman, Shane Leonard, and Andreas Paepcke. 2015. YouEDU: addressing confusion in MOOC discussion forums by recommending instructional video clips. (2015).

[2]https://radimrehurek.com/gensim/
[3]https://dumps.wikimedia.org/enwiki/

[2] Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*. 543–551.

[3] Christopher G Brinton and Mung Chiang. 2015. MOOC performance prediction via clickstream data and social learning networks. In *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2299–2307.

[4] Jason Chuang, Christopher D Manning, and Jeffrey Heer. 2012. "Without the clutter of unimportant words": Descriptive keyphrases for text visualization. *ACM Transactions on Computer-Human Interaction (TOCHI)* 19, 3 (2012), 19.

[5] Carleton Coffrin, Linda Corrin, Paula de Barba, and Gregor Kennedy. 2014. Visualizing patterns of student engagement and performance in MOOCs. In *Proceedings of the fourth international conference on learning analytics and knowledge*. ACM, 83–92.

[6] Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1105–1115.

[7] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[8] Sujatha Das Gollapalli and Cornelia Caragea. 2014. Extracting keyphrases from research papers using citation networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[9] Philip J Guo and Katharina Reinecke. 2014. Demographic differences in how students navigate through MOOCs. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 21–30.

[10] Kazi Saidul Hasan and Vincent Ng. 2010. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 365–373.

[11] Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers)*, Vol. 1. 1262–1273.

[12] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 216–223.

[13] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 366–376.

[14] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 257–266.

[15] Patrice Lopez and Laurent Romary. 2010. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th international workshop on semantic evaluation*. Association for Computational Linguistics, 248–251.

[16] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010

[17] Juan Martinez-Romo, Lourdes Araujo, and Andres Duque Fernandez. 2016. SemGraph: Extracting keyphrases following a novel semantic graph-based approach. *Journal of the Association for Information Science and Technology* 67, 1 (2016), 71–82.

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

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[21] Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *International conference on Asian digital libraries*. Springer, 317–326.

[22] Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 875–884.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[24] Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information retrieval* 2, 4 (2000), 303–336.

[25] Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge.. In *AAAI*, Vol. 8. 855–860.

[26] Rui Wang, Wei Liu, and Chris McDonald. 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software Engineering Research Conference*, Vol. 39.

[27] Ian H Witten and Olena Medelyan. 2006. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06)*. IEEE, 296–297.

[28] Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. KEA: Practical Automated Keyphrase Extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*. IGI Global, 129–152.

[29] Kuldeep Yadav, Ankit Gandhi, Arijit Biswas, Kundan Shrivastava, Saurabh Srivastava, and Om Deshmukh. 2016. Vizig: Anchor points based non-linear navigation and summarization in educational videos. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 407–418.