Unsupervised Keyword Extraction for Hashtag Recommendation in Social Media

Behafarid Mohammad Jafari, Xiao Luo, Ali Jafari
Purdue School of Engineering and Technology
Indiana University-Purdue University Indianapolis
799 West Michigan Street
Indianapolis, IN 46202
behmoham@iu.edu, luo25@iupui.edu, jafari@iu.edu

Abstract

Hashtag recommendation aims to suggest hashtags to users to annotate and describe the key information in the text, or categorize their posts. In recent years, several hashtag recommendation methods are proposed and developed to speed up processing of the texts and quickly find out the critical phrases. The methods use different approaches and techniques to obtain critical information from a large amount of data. This paper investigates the efficiency of unsupervised keyword extraction methods for hashtag recommendation. To do so, well-known unsupervised keyword extraction methods are applied to three real-world datasets including a new dataset containing texts of user-generated posts on a social learning platform. Experimental evaluations demonstrate that statistical methods performs newer methods including graph-based and embedding-based approaches in generating hashtags for long text, whereas the embedding-based approaches works better on generating hashtags for short texts. As a consequence, it can be concluded that unsupervised keyword extraction models can be adapted for hashtag recommendation when the social platform is new or there is no existing data to develop dedicated supervised learning models.

Introduction

The significant growth of text documents on the Internet does not allow users to read content in detail anymore. Assigning relevant tags to the texts which contain important words or short phrases helps to understand the topics faster. With the rapid developments of online social media, social microblogging platforms have become hugely popular providing users around the world with an opportunity to communicate and share their interests. Users generate texts to broadcast recent news, academic achievements, a discovery, etc., to their followers. Users usually assign tags named hashtags to the content they generate on these platforms. These hashtags are types of labels or metadata tags to briefly explain the point of a text or to make it easier to find messages with a specific theme or content [Gong and Zhang2016].

However, not all user-generated content on social media is hashtagged. Recommending hashtags for published text on social media becomes a popular research topic in recent years. Hashtag recommendation can be classified as a natural language processing (NLP) task that enriches a document with key contents that are explicitly or implicitly mentioned in the text. Existing research has studied various models using language features and models [Liu, Chen, and Sun2011]. The collaborative filtering [Wang et al.2014] used often in the recommender system has also been studied. The generative models [Gong, Zhang, and Huang2018] based on textual and social information have also been investigated. Although topic modeling [Godin et al.2013] has been applied on long microblogs to automate the hashtag recommendation, most of the existing models are supervised learning models including the most recent ones based on language embedding models [Kaviani and Rahmani2020, Cantini et al.2021, Kumar et al.2021]. That means the hashtag recommendation task is modeled as a classification problem. Finding a reliable and rich training data set is a great challenge in supervised approaches. Moreover, the trained models in one domain might not be transferred to another domain.

CourseNetworking(CN)[1] is a social learning platform that allows users to publish posts visible to the public or a private group specified by the user [Mohammad Jafari, Zhao, and Jafari2022]. Users on CN usually post about their academic projects, article abstracts, course information, scientific events, etc. They have access to all public posts, and they can view private posts shared with them on their home feed. For a new social network platform like CourseNetworking (CN) – a social learning network, the heuristics rules used in the existing supervised learning models are not applicable. Thus, we propose to investigate unsupervised methods for hashtag recommendation.

This paper aims to extensively evaluate the unsupervised keyword extraction methods for hashtag recommendation. The contributions are: (1) we created a new benchmark dataset (to be released to Github after the paper acceptance) extracted from a social learning platform; (2) and compared twelve unsupervised keyword extraction methods on this new dataset in addition to two Twitter datasets.

Unsupervised Keywords Extraction for Hashtag Recommendation

Given a short text, we assume the labeled hashtags need to represent the key content that the user wants to emphasize. Unsupervised keyword extraction methods first generate a

[1] https://www.thecn.com

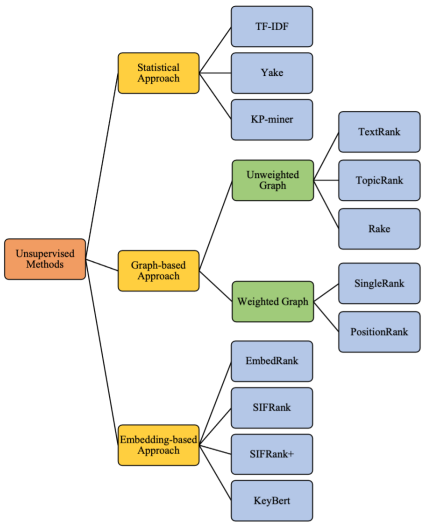

Figure 1: Overview of Unsupervised Keyword Extraction

set of candidates based on the text, then rank the candidates based on their importance to the text. The selected top n candidates are the keywords for the text. This process can be used to generate top n hashtags for recommendation.

The unsupervised keywords can be categorized into three major folds: statistical-based approaches, graph-based approaches, and embedding-based approaches. Figure 1 summarizes the methods included in this research.

Statistical Approaches

Statistical approaches include simple methods that are mainly language and domain-independent and do not require any training data. Statistical approaches use statistical features such as word frequency, n-gram feature, position of the word in the text, and document grammar [Sun et al.2020b] to analyze candidate keyword features and to extract the most important one. They have been widely used and compared to other methods [Sun et al.2020a].

TF-IDF [Salton, Yang, and Yu1975] measures the importance of a word to a document by calculating the TF-IDF value for each candidate. The Term Frequency (TF) represents the frequency a word in a single document, and the Inverse Document Frequency (IDF) represents how frequent the word T has appeared in the whole collection of documents. High score means high term frequency in the given document and a low frequency in the whole set.

KP-Miner [El-beltagy2006] adds two new conditions while selecting candidate words to TF-IDF method. The first condition is the least allowable seen frequency (lasf) factor meaning that a word can be cosidered as a candidate only if it appears in the document more than a given number of times. There is a given threshold position for the second condition. Words cannot be considered as candidates unless they appear in the text before the given cutoff.

Yake [Campos et al.2020] uses five features to score the candidates. It gives higher priority to the capitalized words and acronyms. This method considers the words that occur at the beginning of the text more important than the ones occurring after since it is assumed that important words in a text tend to appear as the very first words. As the third feature, yake calculates the word frequency normalized by 1-standard deviation from the mean. The model then computes how related each word is to the content by counting the times a word occurs in the text with different terms in the left or right of the candidate word. The important words in a content usually occur frequently with the same words next to them. The last feature considers the candidate words than occur more often in different sentences more important. A mathematical combination of all five features makes the final score of the words.

Graph-based Approaches

Graph-based methods are common keyword extraction models [Mothe, Ramiandrisoa, and Rasolomanana2018]. Graphs are mathematical models that enable the effective and efficient exploration of relationships and structural information [Beliga, Meštrović, and Martinčić-Ipšić2015] hidden in the text. The well-known graph-based approaches can be further categorized into weighted and unweighted graph-based approaches.

TextRank [Mihalcea and Tarau2004] uses PageRank for keyphrase extraction task . The method filters the candidate words based on their part of speech and links them according to the co-occurence relationships. If two words are within a moving window of a parametrized size, they would be considered connected while generating a directed graph where nodes represent the candidate words. PageRank is then applied to compute the words' scores. The score of a keyphrase is the sum of scores of each word in that term.

TopicRank [Daille2013] is an improvement of the TextRank . However, while TextRank uses candidate words as graph vertices, TopicRank uses topics as graph nodes. The document is divided to multiple topics using hierarchical agglomerative clustering. A graph-based ranking model derived from PageRank is then applied to score each topic. Keyphrases are finally generated from each of the topranked topics.

Rake [Rose et al.2010] creates a word degree matrix based on the word co-occurrences in the text. Degree score for a word is then calculated as the sum of the number of co-occurrences divided by their occurrence frequency. The sum of degree score of each word in a phrase then gives the total score of a keyword or a keyphrase. Finally, the candidates are ranked based on their score. NLTK is used to implement this model in Python.

SingleRank [Wan and Xiao2008] works on the basis of the TextRank. However, there are two major differences between these two methods. First of all, unlike TextRank, SingleRank generates a weighted graph meaning that some weight is assigned to each edge based on the distance between the two words within a predefined window. The predefined window size in the SingleRank method is 10 so SingleRank can consider n-grams with an n>2. The second difference is in the number Uni-gram the algorithm retains as potential keywords after running the PageRank method.

While TextRank retains the top one-third of the nodes based on their scores, SingleRank retains all the uni-grams.

PositionRank [Florescu and Caragea2017] first selects Nouns and adjectives as candidate words and generates a weighted graph with candidates as vertices. Links between nodes represents co-occurance of two words within a window of contiguous tokens and co-occurance count within the same window computes the weights. The model then computes a position biased PageRank score for each candidate and recursively computes the scores until the difference between two consecutive iterations very small or a great number of iterations is reached. Word scores in each keyphrase are summed to get the keyphrase score and rank the keyphrases.

TextRank, TopicRank, and Rake are unweighted graph-based algorithms, whereas SingleRank and PositionRank are weight graph-based algorithms.

Embedding-based Approaches

While graph-based approaches have shown promising results, embedding-based approaches have also gained attention in recent years. These methods represent words as high-dimensional vectors, which capture their semantic and syntactic properties. Embedding-based approaches have shown promising results in various studies [Nikzad-Khasmakhi et al.2021] [Wang, Liu, and McDonald2015]. Well-known graph-based approaches include KeyBert, EmbedRank, SIFRank, and SIFRank+ which are explained below.

KeyBert [Grootendorst2020] is a method that benefits from BERT-embeddings. To get a document-level representation, document embeddings are extracted with BERT and word embeddings are then extracted for N-gram phrases. Finally, the model uses a simple cosine similarity to find the most relevant keyphrases.

EmbedRank [Bennani-Smires et al.2018] requires a single document, rather than a corpus. It first extracts candidate phrases from the text, based on part-of-speech sequences. The algorithm then benefits from document embeddings and embeds candidate phrases and the original document in same high-dimensional vector space. Finally, it ranks the candidates based on how close their semantic embeddings are to the embeddings of the original document to lect the output keyphrases.

SIFRank [Sun et al.2020b] calculates phrase and document embeddings. SIFRank combines sentence embedding model SIF and autoregressive pre-trained language model ELMo. It also uses position-biased weights for candidate phrases to improve the relevancy of the keyphrases in long documents.

SIFRank+ [Sun et al.2020b] is the upgraded version of SIFRank using position-biased weight to improve the performance on longer documents.

Social Media Datasets

One New Dataset: In this paper, we introduced a new dataset – "CN Posts Dataset" provided by our sponsor. CourseNetworking (CN) is a social learning and education platform where users can create ePortfolios, join a course, create private or public posts, find friends, etc. For this research, we collected 300 user-generated posts from CN platform. The topics vary from the summary of a book to comparing former US presidents. Each post has 2 to 5 hashtags. It must be noted that no personal or private data is included in this dataset.

Two Existing Datasets: Other than the CN Posts, we include two public real-world datasets: Twitter News and Twitter Covid-19 datasets. Both are collected through Twitter API. Twitter News Dataset consists of 10000 English tweets regarding recent news in June 2022. Twitter Covid-19 Dataset has 6000 English tweets regarding Covid-19. All tweets have at least two hashtags. The dataset will be available to the public after the paper is accepted. Table 1 summarizes the datasets. The 'Ratio' means the portion of the hashtags that can be found within each text. For the CN dataset, all hashtags are found within the text. However, this ratio is lower for Twitter datasets. This tells that more than half of the hashtags contain words that does not appear in the text for Covid dataset and it is even more for Newsdata.

Evaluation and Results

All three datasets were preprocessed by removing URLs, emojis, numerical characters, punctuation characters such as periods and question marks, and stop words. Then, Lemmatization was applied to the words.

To evaluate the models, both model-generated hashtags and the original hashtags are stemmed and then compared. Precision (P), Recall (R), and F1 score were used as evaluation metrics. For all datasets, the minimum number of hashtags for each text is 2, and the maximum average number of hashtags for each text is 3.55. We applied the unsupervised keyword extractions to identify the top 3 and 5 candidates and compared them against the ground truth hashtags. The results are shown in Table 2. It clearly shows that for the CN dataset, Yake performs better than the other models. For the Twitter Covid-19 dataset, the results imply that KeyBert has the best performance amongst others, whereas Yake performs slightly lower than KeyBert. For the Twitter News dataset, it concludes that EmbedRank outperforms other methods. Considering all three datasets, Yake shows an advantage over the others.

Comparing the methods in the three categories, statistical-based models perform better than graph-based and embedding-based models on long texts, whereas the embedding-based models seem to perform slightly better than other models on short texts, such as tweets. We think the main reason is that the graph-based or embedding-based models often consider the relations between the candidates and the whole text, hence, it works well on social media texts that are often short text comparing to the abstracts or documents of scientific articles. Whereas for long texts, such as blogs, they often include multiple complete sentences. The statistical-based models consider the static features of the words within the sentences, such as the position of the words, etc. The statistical-based models work better for long-text hashtag recommendations.

Table 1: Evaluation Datasets

heightDataset	Type	#Docs	#Avg Char/Text	#Avg Token/Text	#Avg Hashtag/Text	Ratio	Labels Type
Twitter Covid-19	Tweets	6000	212	34	2.37	43.5%	User Generated
Twitter News	Tweets	10000	191	28	3.8	35.1%	User Generated
CN Posts	Learning Posts	300	930	160	3.55	100%	Author Assigned

Table 2: Model Performance on all datasets

Recommended n Hashtags	Algorithms	Twitter News Data			Twitter Covid Data			CN Dataset		
		P	R	F1	P	R	F1	P	R	F1
3	Statistical Approach									
	TF-IDF	6.66	6.43	6.54	4.77	6.80	5.60	8.02	7.35	7.66
	KP-miner	3.55	5.68	4.36	3.63	5.89	4.48	23.57	10.58	14.6
	Yake	11.66	10.30	10.94	9.55	13.55	11.20	25.64	22.97	**24.22**
	Graph-based Approach									
	TextRank	8.99	4.87	6.32	7.66	7.52	7.58	6.57	6.27	6.42
	TopicRank	1.66	1.47	1.56	3.05	4.06	3.48	11.81	10.32	11.02
	SingleRank	1.00	0.89	0.94	1.44	1.80	1.60	0.22	0.19	0.2
	PositionRank	3.50	1.37	1.96	3.83	5.17	4.40	12.82	11.61	12.18
	Rake	0.66	0.58	0.62	0.49	0.24	0.32	2	1.4	1.64
	Embedding-based Approach									
	EmbedRank	11.91	13.25	**12.54**	6.68	9.13	7.72	15.94	14.39	15.12
	SIFRank	0.21	0.37	0.26	0.26	0.54	0.34	1.22	1.05	1.12
	SIFRank+	0.29	0.46	0.34	0.31	0.63	0.40	7.35	6.75	7.04
	KeyBert	3.66	4.30	3.94	9.33	15.94	**11.76**	15.38	14.02	14.66
5	Statistical Approach									
	TF-IDF	5.59	8.55	6.76	4.06	9.76	5.74	7.96	11.8	9.5
	KP-miner	4.12	6.23	4.96	4.52	6.67	5.38	23.69	11.02	15.04
	Yake	11.05	18.13	13.72	8.44	20.79	12.00	21.07	31	**25.08**
	Graph-based Approach									
	TextRank	8.83	5.36	6.62	12.97	8.38	8.04	10.11	15.33	12.18
	TopicRank	2.06	2.46	2.24	2.94	5.23	3.76	8.44	12.06	9.92
	SingleRank	1.46	1.84	1.62	2.07	3.87	2.68	1.53	2.25	1.82
	PositionRank	3.83	2.82	3.24	3.90	6.87	4.98	12.44	18.5	14.86
	Rake	1.38	2.58	1.80	0.83	1.13	0.96	2.1	1.66	1.86
	Embedding-based Approach									
	EmbedRank	11.30	19.86	**14.40**	7.55	17.87	10.62	12.91	19.41	15.5
	SIFRank	0.32	0.45	0.36	0.39	0.62	0.48	2.4	3.83	2.96
	SIFRank+	0.36	0.53	0.42	0.47	0.78	0.58	6.42	9.76	7.74
	KeyBert	4.40	6.31	5.18	9.03	24.62	**13.20**	14.04	21.31	16.92

Table 3: Models performance on Unigram and Bigram hashtags on CN dataset

Type of Hashtag	TF-IDF	KP-Miner	Yake	TextcRank	TopicRank	SingleRank	PositionRank	Rake	EmbedRank	SIFRank	SIFRank+	KeyBert
Unigram	12.92	21.24	**31.36**	17.62	12.92	1.08	17.12	3.54	21.80	3.74	11.26	15.98
Bigram	10.88	3.06	2.58	14.62	11.46	6.12	**24.74**	0.00	3.08	5.96	13.60	19.16

Table 4: Case Study

Dataset	Original Text
Twitter Covid-19	Funny — Italian man wears isolation disk to keep coronavirus at bay. Avoid public gatherings COVID
CN Posts	Hey! Calling all drop-of-blood donors!raising-hands Do you know you can save 3 lives by donating a bag of drop-of-blood? Yes, up to 3 lives! And this brings the meaning behind the event name, "Blood Donating, World Beating: You Save Lives!". By donatingblood, you keep the world's heart beatingglobe-showing-Americasred-heart
Twitter News	Let's get a grip on ourselves over the latest flag news. Out of over 50 countries and territories in the Commonwealth only 4 countries have the "redJack" as part of their national flag. Us and Fiji, Tuvalu and Aotearoa New Zealand. blueAustralia, blueadambandt

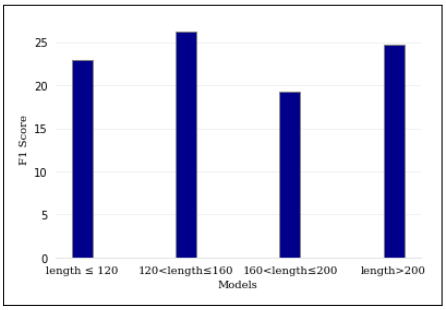

Figure 2: Length Analysis on CN Data

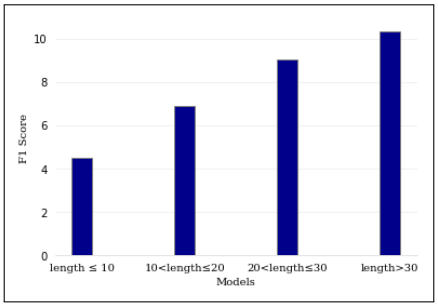

Figure 3: Length Analysis on News Data

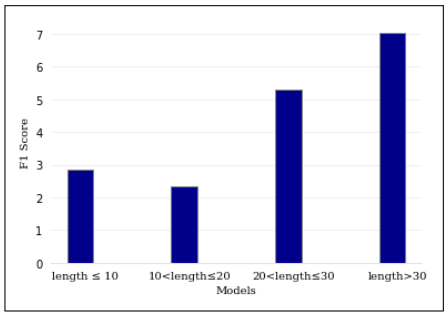

Figure 4: Length Analysis on Covid Data

Model Analysis

Performance on Unigram and Bigram Hashtags can consist of one or more words that are concatenated into one, such as #Covid, #LearningOutcomes, etc. Hence, we investigated the model performances on unigram and bigram hashtag recommendation. Since the CN posts dataset has more hashtags with two or more words, we used it to demonstrate the performance differences. Table 3 shows the F1 score comparison of unigram and bigram hashtag recommendations when k is set to 5. For unigrams, Yake significantly outperforms the other models. This is consistent with the overall performance comparison. The reason is that the majority of the hashtags are unigrams. For bigrams, PositionRank performs the best. We think the reason is that PositionRank considers the co-occurrence of two words within a window of contiguous tokens.

Performance on Various Lengths of the Text The unsupervised keyword extraction methods can perform differently on text with different lengths based on the number of words. Figures 2, 3, and 4 illustrate the performance of Yake on each data set with text of various lengths. For the two Twitter datasets, the majority of the texts have less than 100 words. Whereas the CN Posts dataset that most of the texts have more than 100 words. The performance figures show that the algorithm performs better on longer text. However, for the CN Posts dataset, when the length increases to more than 160 words, the increase in the performance is not significant.

Case Study

In this section, we explore false predictions using examples which can shred the light on how models should be designed in the future. Table 4 shows samples from the datasets that Yake could not predict some of the hashtags when k is set to be 3. The words that are highlighted in green are the true positive ones; the words that are highlighted in blue are the false negatives; and the words that are highlighted in red color are the false positive ones.

We found that one limitation of unsupervised keyword extraction methods is that they rely on the original content. The ones that are not in the content can't be identified, such as 'COVID' which is tagged in many tweets, although the tweet content does not have the term 'COVID'. Upon analyzing the texts and predictions for the CN dataset, it became apparent that due to its nature as a social learning platform, the CN dataset contains a diverse range of words, emojis, and characters. Users often use emojis instead of language to emphasize the content, and algorithms may miss important information by not translating these emojis into text. To address this challenge, an algorithm needs to be designed that takes into account specific characters, emojis, and domain-specific terms to capture the crucial information conveyed by these tokens.

Although algorithm Yake works well on CN data set, it predicts keywords by giving more weight to the capitalized words and phrases. This could generate false positive predictions. For example, in tweets of Twitter news, some words are capitalized without being hashtags. On the other hand, it might give high score to capitalized word, such as "LLP", etc, which are not important.

Conclusion and Future Work

Indexing, searching, and classifying content can be incredibly time-consuming. Hashtags serve as labels for user-generated text, succinctly summarizing the content in just a few words. This paper examines the efficiency of unsupervised keyword extraction algorithms for hashtag recommendation. The evaluation and results indicate that unsupervised keyword extraction approaches are promising in finding potential hashtags in text, even in the absence of training data.

Future developments in this field include (1) the creation of methods to recommend hashtags that are not part of the text, utilizing other NLP techniques such as short text summarization, and (2) the introduction of novel few-shot learning algorithms to recommend hashtags based on a limited number of known examples within the domain.

References

Beliga, S.; Meštrović, A.; and Martinčić-Ipšić, S. 2015. An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences* 39(1):1–20.

Bennani-Smires, K.; Musat, C. C.; Hossmann, A.; Baeriswyl, M.; and Jaggi, M. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *CoNLL*.

Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; and Jatowt, A. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences* 509:257–289.

Cantini, R.; Marozzo, F.; Bruno, G.; and Trunfio, P. 2021. Learning sentence-to-hashtags semantic mapping for hashtag recommendation on microblogs. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16(2):1–26.

Daille, A. B. F. B. B. 2013. Graph-based topic ranking for keyphrase extraction.(2013).

El-beltagy, S. R. 2006. Kp-miner: A simple system for effective keyphrase extraction. In *2006 Innovations in Information Technology*, 1–5.

Florescu, C., and Caragea, C. 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)*, 1105–1115.

Godin, F.; Slavkovikj, V.; De Neve, W.; Schrauwen, B.; and Van de Walle, R. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd international conference on World Wide Web*, 593–596.

Gong, Y., and Zhang, Q. 2016. Hashtag recommendation using attention-based convolutional neural network. In *IJCAI*, 2782–2788.

Gong, Y.; Zhang, Q.; and Huang, X. 2018. Hashtag recommendation for multimodal microblog posts. *Neurocomputing* 272:170–177.

Grootendorst, M. 2020. Keybert: Minimal keyword extraction with bert.

Kaviani, M., and Rahmani, H. 2020. Emhash: Hashtag recommendation using neural network based on bert embedding. In *2020 6th International Conference on Web Research (ICWR)*, 113–118. IEEE.

Kumar, N.; Baskaran, E.; Konjengbam, A.; and Singh, M. 2021. Hashtag recommendation for short social media texts using word-embeddings and external knowledge. *Knowledge and Information Systems* 63(1):175–198.

Liu, Z.; Chen, X.; and Sun, M. 2011. A simple word trigger method for social tag suggestion. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, 1577–1588.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411.

Mohammad Jafari, B.; Zhao, M.; and Jafari, A. 2022. Rumi: An intelligent agent enhancing learning management systems using machine learning techniques. *Journal of Software Engineering and Applications* 15:325–343.

Mothe, J.; Ramiandrisoa, F.; and Rasolomanana, M. 2018. Automatic keyphrase extraction using graph-based methods. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 728–730.

Nikzad-Khasmakhi, N.; Feizi-Derakhshi, M.-R.; Asgari-Chenaghlu, M.; Balafar, M.-A.; Feizi-Derakhshi, A.-R.; Rahkar-Farshi, T.; Ramezani, M.; Jahanbakhsh-Nagadeh, Z.; Zafarani-Moattar, E.; and Ranjbar-Khadivi, M. 2021. Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding. *arXiv preprint arXiv:2106.04939*.

Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory* 1(1-20):10–1002.

Salton, G.; Yang, C.-S.; and Yu, C. T. 1975. A theory of term importance in automatic text analysis. *Journal of the American society for Information Science* 26(1):33–44.

Sun, C.; Hu, L.; Li, S.; Li, T.; Li, H.; and Chi, L. 2020a. A review of unsupervised keyphrase extraction methods using within-collection resources. *Symmetry* 12(11):1864.

Sun, Y.; Qiu, H.; Zheng, Y.; Wang, Z.; and Zhang, C. 2020b. Sifrank: a new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access* 8:10896–10906.

Wan, X., and Xiao, J. 2008. Collabrank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 969–976.

Wang, Y.; Qu, J.; Liu, J.; Chen, J.; and Huang, Y. 2014. What to tag your microblog: Hashtag recommendation based on topic analysis and collaborative filtering. In *Asia-Pacific web conference*, 610–618. Springer.

Wang, R.; Liu, W.; and McDonald, C. 2015. Using word embeddings to enhance keyword identification for scientific publications. In *Databases Theory and Applications: 26th Australasian Database Conference, ADC 2015, Melbourne, VIC, Australia, June 4-7, 2015. Proceedings 26*, 257–268. Springer.