The 2016 Conference on Computational Linguistics and Speech Processing
ROCLING 2016, pp. 402-411

Generation of Conceptual-Level Text Cloud with Graph Diffusion

Ying-Chun Lin, Po-An Yang, Yen-Kuan Lee and Kun-Ta Chuang

Dept. of Computer Science and Information Engineering

National Cheng Kung University

yclin@netdb.csie.ncku.edu.tw, yangpoan@gmail.com, yenkuanlee@gmail.com,

ktchuang@mail.ncku.edu.tw

Abstract

In this paper, we explore a novel framework to generate a well-known text cloud visualization with the conceptual sense. The traditional text cloud is usually generated according to the word occurrence, possibly including the idf-based concept for word weight. The solution is applicable for the long articles. However, for a set of short sentences such as daily news titles, we cannot easily understand the weight of each keyword and its importance to users since the idf value and occurrence in short sentences are difficult to be both well discriminative. In this paper, we propose a graph-based diffusion model to generate conceptual level keyword cloud. We utilize the RDF-based Wikipedia word relation and apply in the Chinese news titles from different news sources. The result shows that our visualization can easily capture the importance concept revealed in a set of news titles.

Keywords: Keyword extraction, Document analysis, Document visualization, Graph-based ranking algorithm

1. INTRODUCTION

As of today, information is ubiquitously generated as the advance of Internet and mobile device. News, movies or books are digitalized to reach more people around the world. For example, Google undertook a Google Books Library Project[1] to scan some old and hard-copy books into its search database, allowing people to query the content in the convenient

[1]http://www.google.com.tw/googlebooks/library/

manner. Information from anywhere and at anytime can be reached in seconds. However, the amount of information needed for each person is significantly small comparing to the all available information. The previous keyword extraction algorithms [5] have been developed for providing concise yet accurate keywords about an article or a piece of news. As such, people can easily acquire sufficient information by merely viewing the keywords.

On the other hand, people can consume less time to grasp the concept of a piece of information simply from the extracted keywords. For example, if we know a piece of news has "AlphaGo", "Lee Sedol" and "Go match" as its keywords, it is easy to know that the news is about the Go match[2] between the 18-time world champion, Lee Sedol, and a computer Go program, AlphaGo. Likewise, if we extract the keywords from a complex system, we can help people understand the system without diving into large amount of information within the system. Take another scenario as an example. To understand the current status in the US, we simply collect the news title of the past few weeks or months and extracted the keywords from those titles. The extracted keywords from the titles may contains "Hillary", "Trump" and "President". It is easy to know that this year is the election year for the US president. Users can easily conclude that the current biggest issue of the US is the presidential election.

However, some important keywords do not always frequently appear in the title of the news. For instance, Pokemon Go[3], a promising mobile game, can be seen in many tech news during June and July, 2016. But Niantic Inc., which is the developer of the game, is seldom mentioned in the news titles. The developer company should be as important as the game when we want to understand the trend in tech at that time. Similarly, the conceptual keywords behind the explicit keywords should be considered when keywords are extracted for any other complex systems. The keyword extraction process is more comprehensive when the conceptual and explicit keywords are both considered.

To extract keywords from a system comprehensively, in this paper, we developed a Knowledge Extracting Framework (KEF). The KEF consists of 5 phases. The primary

[2]https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol

[3]http://www.pokemongo.com/

goal of the KEF is to extract keywords from the given information about the interested system, such as the current status of a country. To resolve the problem of conceptual keyword extraction, we outsource to a well-established knowledge graph built from Wikipedia[4]. We also refer to the technology of graph diffusion to evaluate the importance of each term, including the conceptual terms and explicit terms used in the documents. As such, some hidden conceptual keywords related to the explicit terms can be found from the Wikipedia knowledge base. After extracting all the keywords, KEF will further visualize the result into a Keyword Cloud (KC) according to their significance to the interested system. The KC helps people understand the concepts or issues in the interested system at a glance.

2. RELATED WORKS

The keyword extraction algorithms are investigated to help readers better understand a single document or a collection of documents by the high-level descriptions. A popular mean to extract keywords for each document from a collection of documents is TF-IDF (Term Frequency - Inverse Document Frequency). Instead of identifying the keywords of a single document, Lee *et al.* re-defined the calculation of the term frequency to find the keywords of a collection of news articles[3]. On the other hand, some works evaluate the keywords based on merely the structure of a single document, such as the co-occurrence in sentences between each terms[7] or the dominance between words by influence interval structure[2].

The graph-based keyword extraction algorithms aim to identify keywords by the graph of words and relations between those words. In [8], Mihalcea *et al.* employ PageRank [9] to find the important words in a document and develop the TextRank framework. In [10], Yang *et al.* add the topic information into each word in the graph of TextRank to add the semantic relation between words. Therefore, some latent relations between words are found by assigning the topic to each word. Another algorithm which considers the topic of each word is Topical PageRank (TPR) proposed by Liu *et al.* [6]. They state that the measurement for the word importance should be separately considered in different topics, as shown in their result, TPR can find the keywords more accurately.

[4]https://en.wikipedia.org/

However, the above algorithms can not solve the problem of conceptual keywords mentioned in Section 1. Because those algorithms find keywords only based on the collected documents. If a hidden conceptual keyword is never mentioned in any document of the collection, it is impossible to consider the conceptual keyword as important information. Then, the extracted keywords may not be comprehensive for understanding an interested system.

3. KNOWLEDGE EXTRACTING FRAMEWORK

We aim to exploit the simplicity and accuracy of keywords to help people understand a complex system quickly and easily. In this paper, "system" refers to various possible paradigms, which can be the current status of a country, the fashion trend or the products of a company. As long as we have the information relevant to the targeted system, Knowledge Extracting Framework (KEF) utilizes the information to generate a Keyword Cloud (KC). As a result, the comprehensive concept about the complex system can be grasped at a glance.

The KEF consists of four phases: keyword extraction, diffusion, significance evaluation and text cloud visualization. $D = \{d_1, d_2, ..., d_n\}$ are the documents related to the targeted system. After keyword extraction phase, $K_e = \{k_{e_1}, k_{e_2}, ..., k_{e_m}\}$ are extracted from D as well as the term frequency of each keyword. To reveal the conceptual information behind K_e, we use a knowledge graph of the RDF format in diffusion phase. The conceptual keywords $K_i = \{k_{i_1}, k_{i_2}, ..., k_{i_l}\}$ can be obtained from the RDF graph. Both K_e and K_i are ranked by the significance scores calculated in significance evaluating phase. Finally, the keywords are visualized in a KC according to the their significance scores.

3.1 Keyword Extraction Phase

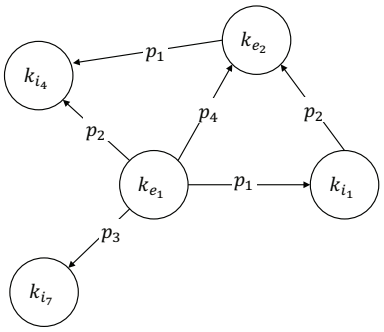
We first transform the sentence into a set of terms. As such, each document in D is represented by a set of terms. We use the tf-idf technique to identify the explicit keywords K_e in D. After filtering the stop words, the overall term frequency $TF(k_{e_x})$ for each $k_{e_x} \in K_e$ is calculated within all documents in D. $TF(k_{e_x})$ can be considered as the

significance score for each keywords. However, the problem of the conceptual terms may appear. Consequently, users cannot easily obtain a comprehensive KC.

3.2 Diffusion Phase

In this phase, we aim to solve the conceptual keyword problem. The conceptual keywords may not be seen or not appear frequently in D. However, the conceptual keywords are as critical as K_e. In following sections, we call the conceptual keywords the implicit keywords K_i.

To find implicit keywords K_i behind K_e, we refer to a well-established RDF graph built from DBpeida[1][4]. DBpeida is a database maintaining all information in Wikipedia. The storing format of the database is n-triple, such as $triple = (subject, predicate, object)$. The subject is related to the object by the predicate. For example, (Pokémon Go, developer, Niantic Inc.) indicates that Pokémon Go is developed by Niantic Inc. A triple is a fact. DBpedia is the database consisting of more than billion facts. These facts can be transformed into the RDF graph. In the RDF graph, the subjects and the objects are the nodes and the predicates are the edge between nodes.



Keywords	Significance Score
k_{e_1}	$TF(k_{e_1}) + TF(k_{e_2}) \times \delta_1$
k_{e_2}	$TF(k_{e_1}) \times \delta_2 + TF(k_{e_2})$
k_{i_1}	$TF(k_{e_1}) \times \delta_1$
k_{i_4}	$TF(k_{e_1}) \times \delta_1$
k_{i_7}	$TF(k_{e_1}) \times \delta_1 + TF(k_{e_1}) \times \delta_2$

Figure 1. An example of RDF graph.

The implicit keywords K_i are found from the RDF graph. Every explicit keyword k_{e_x} is a node in the RDF graph. The implicit keywords are those nodes found by the 2-step propagation. They are the neighbors of k_{e_x}, represented as $N(k_{e_x})$, and the neighbors of the neighbors, represented as $\bigcup_{k_i \in N(k_{e_x})} N(k_i)$. As shown in Figure 1, the implicit keyword found by k_{e_1} is k_{i_1}, k_{i_4}, k_{i_7} and k_{e_2}. p_1, p_2, p_3 and p_4 are the predicates. The calculation of significance score is discussed in Section 3.3.

3.3 Significance Evaluating Phase

To determine the importance of each keywords, a significance score is given to each K_e and K_i in this phase. For an explicit keywords, the term frequency $TF(k_{e_x})$ is calculated in the keyword extraction phase. The $TF(k_{e_x})$ can be used as a reference score for each keywords. We define the significance score of a keyword as

$$\text{significance score}(k) = \begin{cases} TF(k) & \text{if } k \in K_e \\ TF(k_{e_x}) \times \delta_l & \text{if } k \in K_i \text{ and } k \in [N(k_{e_x}) \text{ or } \bigcup_{k_i \in N(k_{e_x})} N(k_i)] \end{cases}$$

The significance score of an implicit keyword given by an explicit keywords is $TF(k_{e_x}) \times \delta_l$, where l is the distance to the explicit keyword. The adjustment of the value δ_l is shown in the Section 4. As a result, the score of each keywords is illustrated in Figure 1. The score given by k_{e_x} is the term frequency for k_{e_x} and the score for the implicit keywords is $TF(k_{e_x}) \times \delta_l$.

An implicit keyword may be the neighbor of several explicit keywords. An explicit keyword could be the implicit keyword for many other explicit keywords. Therefore, the significance score is accumulated each time when a node in the RDF graph is identified as an implicit keyword.

3.4 Visualization Phase

To help people understand the targeted system quickly and easily, we employ a visualization technique considering both the keywords and their significance - the word cloud. The generated word cloud is called the Knowledge Cloud (KC). In the KC, the top-k keywords are selected according to their scores. The selected keywords are scattered in the KC. If a keyword has higher significance score, its font size is larger.

4. EXPERIMENTAL STUDIES

In this section, we discuss the performance of the KEF framework. To illustrate the simplicity and the accuracy with keywords in a KC, we use the status in Taiwan for each

month as the targeted system. Further, we adjust the decay parameter δ_l to investigate how the value for δ_l can affect the final KC of the targeted system.

4.1 Experimental Setup

We use the news titles for each day in Taiwan as a mean to understand the status in Taiwan every month. The news titles are obtained from a website called NewsDiff[5] and they are published from September, 2013 to June, 2016. Specifically, six major Internet news sources, such as Chinatimes and UDN news, are included. In our experimental studies, only news titles are utilized since the news contents are too noisy. In average, there are 134,395 news generated in each month.

We are interested in showing the status in Taiwan in September 2014, in short 2014 September status. At that time, a series of sit-in street protest, called Umbrella Revolution happened in Hong Kong[6]. The students in Hong Kong led a strike against the decision regarding to the reform of Hong Kong electoral system. The event caught the attention of people in Taiwan because a similar protest, namely Sunflower Student Movement, happened in Taiwan six months earlier. Therefore, the keywords of 2014 September status are highly related to the Umbrella Revolution. Therefore, we compare different methods and observe that whether the keywords related to the 2014 Hong Kong protest appear.

The final result in the KC relies on how to calculate the significance score. The significance score reflects the important concepts happening in 2014 September. To demonstrate that the importance of finding implicit keywords, we compare our method, i.e., KEF, with the Term Frequency (TF). In TF, the frequency of keywords is counted by the number of occurrence in the news titles.

Additionally, the decay parameter δ_l affects the results in the KC as well. The value of δ_1 and δ_2 are set to 1 in the $KEF_{uniform}$, which means the importance of the implicit keywords is the same as the related explicit keywords. The $KEF_{hierachy}$ reduces the significance score of the implicit keywords according to an exponential function $\delta_1 = \lambda e^{-\lambda l}$.

[5]http://newsdiff.g0v.ronny.tw/

[6]https://en.wikipedia.org/wiki/2014_Hong_Kong_protests

4.2 Performance Evaluation

Figure 2. The 2016 February status is generated by the TF method.

In Figure 2, we observe that the words related to the Umbrella Revolution, such as "Mainland China"(大陸), "Hong Kong"(香港) and "Student"(學生), are in the KC generated by the TF method. However, the related information about the event is not observed in the KC. On the other hand, the KC generated by the $KEF_{hierachy}$ has richer information. For example, the Hong Kong protest caused a severe slump in Hong Kong tourism(香港旅遊業). Sen-Hong Yang (楊憲宏) who is a journalist and human rights activist made many comments about the protest. Through the KC of the $KEF_{hierachy}$, users can have a comprehensive understanding about the most interested issue of 2014 September status.

Figure 3. The 2016 February status is generated by the $KEF_{hierachy}$ method.

In Figure 3 and 4, the effect of the value of δ_1 can be observed. For both $KEF_{hierachy}$ and $KEF_{uniform}$, we find the neighbors of k_{e_x} within distance 2 as the implicit keywords. The $KEF_{uniform}$ gives the same significance score to the implicit keywords. The $KEF_{hierachy}$ reduces the significance of the implicit keywords according to the distance to k_{e_x}. The λ value of the exponential function is set to 1.

Comparing to the KC in Figure 2 and 3, the keywords in this Figure 4 are less relevant to 2014 September status. Instead, the KC is full of many hub keywords, which is the keywords related to many explicit keywords. Because the significance score is added by each k_{e_x} equally, the hub keywords can have higher score easily. The effect of δ_1 can be observed through Figure 3 and 4.

Figure 4. The 2016 February status is generated by the $KEF_{uniform}$ method.

5. CONCLUSIONS

In this paper, we aim to generate a conceptual-level KC of an interested system. The generated KC can help people understand a complex system at a glance. We argue that, when the importance of a keyword is weighted, we should consider both term frequency and the conceptual relation between keywords. The proposed framework, KEF, utilizes RDF-based word relation graph to find the hidden relation between keywords. In the significance evaluation phase of the KEF framework, the significance of the keywords is calculated not only based on the term frequency but also the relations between keywords. The experimental results demonstrated that the KEF framework can accurately generate a comprehensive KC for the status in Taiwan in September 2014e. In the future, we plan to apply KEF to different systems, such as tech trend detection. Furthermore, a generalized KEF could be devised as a general-purpose service to help people easily understand the concepts of their interested systems.

ACKNOWLEDGMENTS

This study is conducted under the "Fundamental Industrial Technology Development Program(4/4)" of the Institute for Information Industry which is subsidized by the Ministry

of Economic Affairs of the Republic of China. In addition, this paper is also supported in part by Ministry of Science and Technology, R.O.C., under Contract 105-2221-E-006-140-MY2.

REFERENCES

[1] C. Bizer, S. Auer, G. Kobilarov, J. Lehmann, and R. Cyganiak. Dbpedia—querying wikipedia like a database. In *Developers track presentation at the 16th international conference on World Wide Web, WWW*, pages 8–12, 2007.

[2] D.-Y. Lee, K.-R. Kim, and H.-G. Cho. *A New Extraction Algorithm for Hierarchical Keyword Using Text Social Network*, pages 903–912. Springer, 2016.

[3] S. Lee and H.-j. Kim. News keyword extraction for topic tracking. In *Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on*, volume 2, pages 554–559. IEEE, 2008.

[4] J. Lehmann, J. Schüppel, and S. Auer. Discovering unknown connections-the dbpedia relationship finder. *CSSW*, 113:99–110, 2007.

[5] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, 2008.

[6] Z. Liu, W. Huang, Y. Zheng, and M. Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 366–376. Association for Computational Linguistics, 2010.

[7] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[8] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[10] K. Yang, Z. Chen, Y. Cai, D. Huang, and H.-f. Leung. Improved automatic keyword extraction given more semantic knowledge. In *International Conference on Database Systems for Advanced Applications*, pages 112–125. Springer, 2016.