Expert Systems With Applications 97 (2018) 163–176

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Document-specific keyphrase candidate search and ranking

Qingren Wang[a,*], Victor S. Sheng[b,*], Xindong Wu[c]

[a] Hefei University of Technology, 193 Tunxi Road, Hefei, Anhui Province 230009 China
[b] University of Central Arkansas, 201 Donaghey Ave., Conway, AR 72035 USA
[c] School of Computing and Informatics, University of Louisiana at Lafayette, 222 James R. Oliver Hall, Lafayette, LA 70504-3694, USA

ARTICLE INFO

Article history:
Received 30 June 2017
Revised 14 December 2017
Accepted 16 December 2017
Available online 16 December 2017

Keywords:
Keyphrase candidate search
Sequential pattern mining
Keyphrase candidate ranking
Entropy

ABSTRACT

This paper proposes an approach KeyRank to extract proper keyphrases from a document in English. It first searches all keyphrase candidates from the document, and then ranks them for selecting top-*N* ones as final keyphrases. Existing studies show that extracting a complete keyphrase candidate set that includes semantic relations in context, and evaluating the effectiveness of each candidate are crucial to extract high quality keyphrases from documents. Based on that words do not repeatedly appear in an effective keyphrase in English, a novel keyphrase candidate search algorithm using sequential pattern mining with gap constraints (called KCSP) is proposed to extract keyphrase candidates for KeyRank. And then an effectiveness evaluation measure pattern frequency with entropy (called PF-H) is proposed for KeyRank to rank these keyphrase candidates. Our experimental results show that KeyRank has better performance. Its first component KCSP is much more efficient than a closely related approach SPMW, and its second component PF-H is an effective evaluation mechanism for ranking keyphrase candidates.[1]

1. Introduction

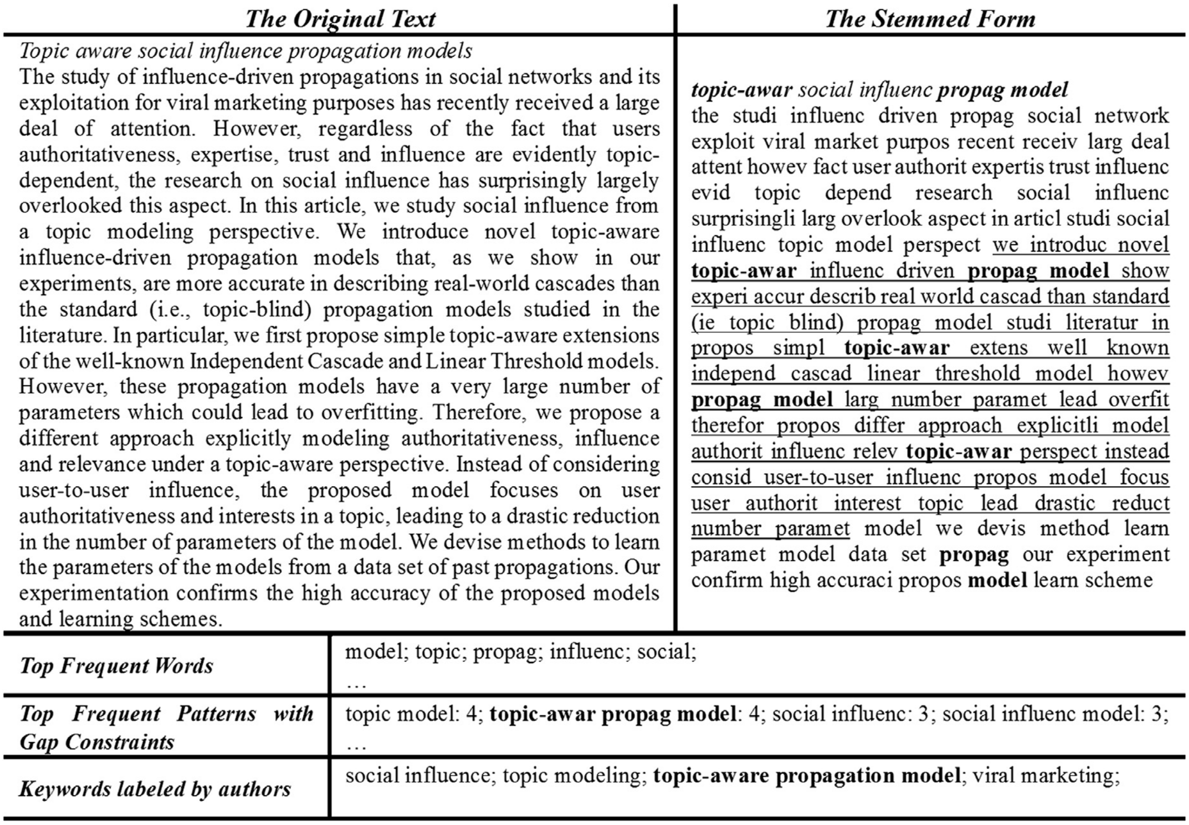
A keyphrase (Liu, Song, Liu, & Wang, 2012) is an ordered list of words that captures the main points discussed in a natural language document. Keyphrases in a document can help understand the main points of this document. Keyphrases have been successfully used in many text mining tasks, such as automatic indexing, topic extraction, document summarization and text categorization, and so on. Due to the importance of keyphrase, many studies have been conducted to extract high quality keyphrases from documents. This is called keyphrase extraction. Existing keyphrase extraction approaches are based on unsupervised learning and supervised learning (Xie, Wu, & Zhu, 2014). They usually contain two components, keyphrase candidate search and keyphrase selection. Keyphrase candidate search is to extract a keyphrase candidate set from a document. After a keyphrase candidate set is extracted, all these approaches conduct keyphrase selection to select proper keyphrases from the keyphrase candidate set using different technologies. For supervised learning based approaches, keyphrase selection is formulated as a classification task, where each candidate is classified as either a keyphrase or a non-keyphrase. For unsupervised learning based approaches, keyphrase selection is to rank keyphrase candidates in terms of a specific measure, and then the top-*N* (*N* is the number of proper keyphrases, specified by users for a document) keyphrase candidates are selected as keyphrases. Studies (Ercan & Cicekli, 2007; Xu, Yang, & Lau, 2010; Haddoudand & Abdeddaim, 2014) showed that semantic relations in context can help improve the performance for keyphrase extraction. Thus, extracting a complete keyphrase candidate set that includes semantic relations in context, and selecting proper keyphrases from the keyphrase candidate set are crucial to extract high quality keyphrases from documents. In this paper, we focus on the two crucial components of keyphrase extraction, keyphrase candidate search and keyphrase selection, and propose an efficient approach for keyphrase extraction (Wang, Sheng, & Wu, 2017).

The original work on keyphrase extraction simply treats single words with high frequency as keyphrase candidates. However, single words do not capture semantic relations in context. Approaches based on single words with high frequency cannot extract a complete keyphrase candidate set that includes semantic relations in context. Some studies considered contiguous frequently-occurring words as keyphrase candidates, such as Kea (Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999). Nevertheless, no matter how many contiguous frequently-occurring words that a keyphrase candidate has, it still ignores some semantic relations in context (Fu, Huang, Sun, Vasilakos, & Yang, 2016; Fu, Ren, Shu, Sun, & Huang, 2016; Li, Li, Yang, & Sun, 2015). Intuitively, single words in a document are the minimum

* Corresponding authors.
E-mail addresses: qingren.wang@mail.hfut.edu.cn (Q. Wang), ssheng@uca.edu (V.S. Sheng), xwu@louisiana.edu (X. Wu).

[1] Our two-page extended abstract is published in AAAI 2017 (Wang, Sheng, & Wu, 2017).

https://doi.org/10.1016/j.eswa.2017.12.031
0957-4174/© 2017 Elsevier Ltd. All rights reserved.



The Original Text	***The Stemmed Form***
Topic aware social influence propagation models The study of influence-driven propagations in social networks and its exploitation for viral marketing purposes has recently received a large deal of attention. However, regardless of the fact that users authoritativeness, expertise, trust and influence are evidently topic-dependent, the research on social influence has surprisingly largely overlooked this aspect. In this article, we study social influence from a topic modeling perspective. We introduce novel topic-aware influence-driven propagation models that, as we show in our experiments, are more accurate in describing real-world cascades than the standard (i.e., topic-blind) propagation models studied in the literature. In particular, we first propose simple topic-aware extensions of the well-known Independent Cascade and Linear Threshold models. However, these propagation models have a very large number of parameters which could lead to overfitting. Therefore, we propose a different approach explicitly modeling authoritativeness, influence and relevance under a topic-aware perspective. Instead of considering user-to-user influence, the proposed model focuses on user authoritativeness and interests in a topic, leading to a drastic reduction in the number of parameters of the model. We devise methods to learn the parameters of the models from a data set of past propagations. Our experimentation confirms the high accuracy of the proposed models and learning schemes.	***topic-awar*** *social influenc* ***propag model*** the studi influenc driven propag social network exploit viral market purpos recent receiv larg deal attent howev fact user authorit expertis trust influenc evid topic depend research social influenc surprisingli larg overlook aspect in articl studi social influenc topic model perspect we introduc novel **topic-awar** influenc driven **propag model** show experi accur describ real world cascad than standard (ie topic blind) propag model studi literatur in propos simpl **topic-awar** extens well known independ cascad linear threshold model howev **propag model** larg number paramet lead overfit therefor propos differ approach explicitli model authorit influenc relev **topic-awar** perspect instead consid user-to-user influenc propos model focus user authorit interest topic lead drastic reduct number paramet model we devis method learn paramet model data set **propag** our experiment confirm high accuraci propos **model** learn scheme

Top Frequent Words	model; topic; propag; influenc; social; …
Top Frequent Patterns with Gap Constraints	topic model: 4; **topic-awar propag model**: 4; social influenc: 3; social influenc model: 3; …
Keywords labeled by authors	social influence; topic modeling; **topic-aware propagation model**; viral marketing;

Fig. 1. Examples of using frequent words vs. wildcard based sequential patterns for keyphrase extraction. (Xie et al., 2014).

meaningful and independent units, and meanwhile a document is an ordered list of words. Therefore, some studies treat the keyphrase candidate search as a task of sequential pattern mining with gap constraints, where single words of documents are viewed as characters of sequences and keyphrase candidates are viewed as patterns. Xie et al. (2014) combined wildcards into sequential pattern mining to search keyphrase candidates from a document (called SPMW), since wildcards can provide gap constraints with a great flexibility for mining patterns to capture semantic relations in the document (Agrawal & Srikant, 1995). Compared with the approaches based on single or contiguous frequently-occurring words, approaches based on sequential pattern mining can discover a richer pattern (keyphrase candidate) set, which helps improve the quality of keyphrase extraction. Note that in this paper a pattern is actually defined as a keyphrase candidate, and keyphrase candidates and patterns are exchangeable respectively since then.

Here we employ and adapt one of the examples provided by Xie et al. (2014) to explain why utilizing sequential pattern mining with gap constraints to search keyphrase candidates is better (see Fig. 1). The example is the title and the abstract of a journal paper published in *Knowledge and Information Systems* (2013). The left part of the second row shows the original title (in italics) and the original abstract, and the right part shows the title and the abstract in a stemmed form. The last row shows four keyphrases labeled by its authors. Among the four keyphrases, the entire string “*topic-aware propagation model*” occurs 0 times in the text (either stemmed or not stemmed). However, sequential pattern mining with gap constraints can extract “*topic-awar propag model*” four times from the text, once in title and three times in the abstract (refer to bold-faced words in the right part of the second row in Fig. 1). Therefore, “*topic-awar propag model*” is extracted as a keyphrase candidate as shown in the fourth row in Fig. 1.

Although sequential pattern mining with gap constraints can extract keyphrase candidates with a higher quality, existing sequential pattern mining based approaches are computational expensive due to two weaknesses. On the one hand, the gap constraints in these approaches play a very important role, but they require users to explicitly specify appropriate gap constraint(s) beforehand. In reality it is often nontrivial and time-consuming for users to provide a proper gap constraint. On the other hand, these approaches need to scan a document multiple times for searching patterns. Repeated document scanning can cause a lot of time overhead, even for a short document. Many studies (Fumarola, Lanotte, Ceci, & Malerba, 2016; Loglisci & Malerba, 2009; Xie et al., 2014) toward closed patterns showed that they preserve information and help keep the computational complexity under control. However, closed patterns do not work well on keyphrase extraction because it neglects to consider the three inherent properties of a pattern to capture a point in a document, especially uncertainty. In this paper, we focus on documents in English and solve the following two issues: 1) reducing the computation time of searching keyphrase candidates, and 2) measuring the probability of a keyphrase candidate to capture a point expressed by its corresponding document.

After having consulted linguists, they confirmed that words do not repeatedly appear in an effective keyphrase in English. Based on their confirmation, we treat keyphrase candidate search as a sequential pattern mining task, and propose a novel keyphrase candidate search algorithm (called KCSP) using sequential pattern mining with gap constraints. KCSP only scans a document once for obtaining every word whose frequency is no less than a given support threshold, and its corresponding appearing positions in the document. Then KCSP generates a corresponding position interval for each word at each appearing position, and treats it as the gap constraint of the word at the current appearing position. Therefore, the gap constraint of a word at an appearing position becomes the inherent property of the word at the current appearing position, rather than an external parameter specified by users.