Topical Word Importance for Fast Keyphrase Extraction

Lucas Sterckx, Thomas Demeester, Johannes Deleu, Chris Develder
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
firstname.lastname@intec.ugent.be

ABSTRACT

We propose an improvement on a state-of-the-art keyphrase extraction algorithm, Topical PageRank (TPR), incorporating topical information from topic models. While the original algorithm requires a random walk for each topic in the topic model being used, ours is independent of the topic model, computing but a single PageRank for each text regardless of the amount of topics in the model. This increases the speed drastically and enables it for use on large collections of text using vast topic models, while not altering performance of the original algorithm.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Automatic Keyphrase Extraction, Topical Keyphrase Extraction

1. INTRODUCTION

Automatic Keyphrase Extraction (AKE) is the task of identifying a set of expressions or noun phrases which concisely represent the content of a given article. Keyphrases have proven useful for various Information Retrieval and Natural Language Processing tasks, such as summarization [1] and contextual advertising on web pages [6]. Currently two types of methods are used: supervised and unsupervised methods. State-of-the-art unsupervised methods transform the input document into a graph representation. Each node in this graph corresponds to a candidate-word and edges connect two candidates occurring within a certain text window. The significance of each node, i.e., word, is computed using a random walk algorithm based on PageRank [4]. The top ranked nodes are then selected to generate keyphrases. TextRank is one of the most well-known examples of a graph-based approach [3]. Recent work has shown that the quality of keyphrases is improved by using topic model information in the graph model. Topical PageRank (TPR) [2] is a variation on the TextRank-algorithm that incorporates topical information by increasing the weight of important topical words based on the topic-document and word-topic distributions generated by a topic model. Experimental results showed that TPR outperforms other existing unsupervised AKE-methods. While TPR is an effective algorithm for the inclusion of topical information from the topic model, it requires a random walk for each topic in the topic model. This approach becomes cumbersome for huge collections of text using large topic models, as PageRank is a computationally intensive algorithm. In this paper we propose a modification of the original TPR algorithm which is equally effective but speeds up the algorithm as many times as the amount of topics in the topic model.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742730.

2. SINGLE-PAGERANK TOPICAL KEYPHRASE EXTRACTION

Topical PageRank, as described in [2], requires a PageRank for each topic separately and boosts the words with high relevance to the corresponding topic. In a word graph each candidate word (i.e., nouns and adjectives) become a vertex in set $\nu = \{w_1, \ldots, w_N\}$. For each candidate w_j, a window of the following words in the given article (typically chosen as 10) is selected and a directed edge from w_j to each word w_i included in the window is created, resulting in a directed graph. Formally, the topic-specific PageRank can be defined as follows:

$$R_z(w_i) = \lambda \cdot \sum_{j:w_j \to w_i} \left(\frac{e(w_j, w_i)}{O(w_j)} \cdot R_z(w_j) \right) + (1-\lambda) \cdot P_z(w_i), \tag{1}$$

where $R_z(w_i)$ is the PageRank score for word w_i in topic z, $e(w_j, w_i)$ is the weight of the edge $(w_j \to w_i)$, the number of outbound edges is $O(w_j) = \sum_{w'} e(w_j, w')$ and λ is a damping factor $\in [0, 1]$ indicating the probability of a random jump to another node. A large $R_z(w)$ indicates a word w that is a good candidate keyword in topic z. The topic specific preference value $P_z(w_i)$ for each word w_i is the probability of arriving at this node after a random jump, thus with the constraint $\sum_{w \in \nu} P_z(w) = 1$ given topic z. In TPR, the best performing value for $P_z(w_i)$ is reported as being the probability that word w_i occurs given topic z, denoted as $P(w_i|z)$. This indicates how much that topic z is focused on word w_i. With the probability of topic z for document d $P(z|d)$, the final ranking score of word w_i in document d is computed as the expected PageRank score over that topic distribution, for a topic model with K topics,

$$R(w_i) = \sum_{z=1}^{K} R_z(w_i) \cdot P(z|d). \tag{2}$$

Adjectives and nouns are then merged into keyphrases and corresponding scores are summed and ranked. Note that original TPR

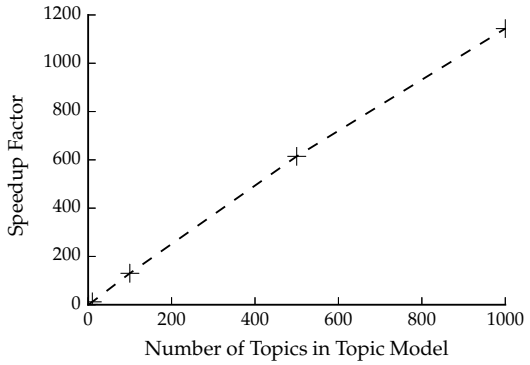

Figure 1: Speed-up with proposed modification

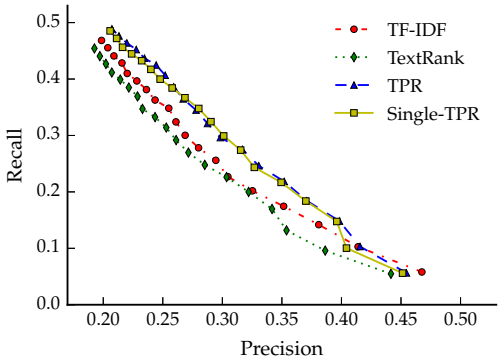

Figure 2: Comparison of the original TPR [2] (indicated 'TPR') with the more efficient single-PageRank TPR (indicated 'single-TPR'), and two baselines, TF-IDF and TextRank [3]

.

requires a PageRank for each topic in the model. Since topic models with a large amount of topics (e.g. $K = 1,000$) are reported to empirically perform best, this requires many computations for each document, especially for long ones. That is, for D documents the total amount of PageRanks for AKE is $K \times D$. We propose an alternative strategy to avoid this large computational cost, by using but a single PageRank per document. We do this by using a single weight-value we call $W(w_i)$ indicating the full topical importance of each word w_i in the PageRank instead of K topic-specific values and summing all results. First, we determine the cosine similarity between the vector of word-topic probabilities $\vec{P}(w_i|Z) = (P(w_i|z_1), \ldots, P(w_i|z_K))$ and the document-topic probabilities of the document, $\vec{P}(Z|d) = (P(z_1|d), \ldots, P(z_k|d))$, to determine the single weight value $W(w_i)$ per word w_i and document d.

$$W(w_i) = \frac{\vec{P}(w_i|Z) \cdot \vec{P}(Z|d)}{\left\|\vec{P}(w_i|Z)\right\| \cdot \left\|\vec{P}(Z|d)\right\|}. \tag{3}$$

This quantity $W(w_i)$ can be considered the 'topical word importance' of word w_i given document d, where the contribution of a particular topic z_k is larger if w_i is an important word for that topic, and the topic is strongly present in the considered document. As a result, the single PageRank $R(w_i)$ becomes

$$R(w_i) = \lambda \cdot \sum_{j:w_j \to w_i} \left(\frac{e(w_j, w_i)}{O(w_j)} \cdot R(w_j) \right) + (1-\lambda) \cdot \frac{W(w_i)}{\sum_{w \in \nu} W(w)}. \tag{4}$$

3. EVALUATION

To detect any change in performance, we use a dataset comprised of news articles built by Wan and Xiao [5], that contains 308 news articles from the 2001 Document Understanding Conference (DUC) summarization-track, with 2,488 manually assigned keyphrases. We create a mapping between the keyphrases in the gold standard and those in the system output using an exact match. We reduce keyphrases to their stems using the Porter-stemmer and use three standard evaluation metrics for AKE: precision, recall, and F1-measure. Other parameters (for the stemmer, tokenizer and PageRank) are identical to those in the original TPR-paper [2]. Figure 1 shows how much our modification speeds up the computation time as compared to the original TPR algorithm for processing of the complete collection of articles. Both approaches are programmed using identical pre-processing functions and PageRank implementations. The graph shows the linear speed up achieved by making the algorithm independent of the amount of topics, and thus constant time. Figure 2 shows precision-recall curves for the original TPR and ours using a single PageRank, using the same topic model of 1,000 topics trained on Wikipedia data (a corpus similar to the one used in the original TPR [2]), and two baselines TF-IDF and TextRank. The effectiveness of our method is close to identical while computation time is reduced by factor $\approx 1/K$ (i.e., 1,000 times faster in this example).

4. CONCLUSION

We propose a more efficient use of topic models for unsupervised keyphrase extraction. Using a single value for topical word importance in a PageRank algorithm based on the cosine similarity between the vector of word-topic probabilities and the document-topic probabilities of the document, we achieve a constant computation time, independent of the topic model being used. We show that this modification does not significantly alter the performance while reducing the computation time by a large margin.

5. REFERENCES

[1] E. D'Avanzo, B. Magnini, and A. Vallin. Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004. In *Proceedings of the 2004 DUC*, 2004.

[2] Z. Liu, W. Huang, Y. Zheng, and M. Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on EMNLP*, pages 366–376, 2010.

[3] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In *Proceedings of the 2004 conference on EMNLP*, 2004.

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

[5] X. Wan and J. Xiao. CollabRank: towards a collaborative approach to single-document keyphrase extraction. *Coling*, pages 969–976, Aug. 2008.

[6] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. WWW 2006, pages 213–222, 2006.