A Position-Biased PageRank Algorithm for Keyphrase Extraction

Corina Florescu and Cornelia Caragea
Computer Science and Engineering
University of North Texas
CorinaFlorescu@my.unt.edu, ccaragea@unt.edu

Abstract

Given the large amounts of online textual documents available these days, e.g., news articles and scientific papers, effective methods for extracting keyphrases, which provide a high-level topic description of a document, are greatly needed. We propose PositionRank, an unsupervised graph-based approach to keyphrase extraction that incorporates information from all positions of a word's occurrences into a biased PageRank to extract keyphrases. Our model obtains remarkable improvements in performance over strong baselines.

Introduction

Keyphrase extraction (KE) is the task of automatically extracting descriptive phrases or concepts that represent the main topics of a document. Keyphrases provide a concise summary of a document and are shown to be rich sources of information in many natural language processing and information retrieval tasks. Due to their importance, many approaches to keyphrase extraction have been proposed in the literature along two lines of research: supervised and unsupervised (Hasan and Ng 2014).

Although supervised approaches typically perform better than unsupervised approaches (Kim et al. 2012; Caragea et al. 2014), the requirement for large human-annotated corpora for each domain of study, has led to significant attention towards the design of unsupervised approaches. Unsupervised keyphrase extraction is formulated as a ranking problem with graph-based ranking techniques being considered state-of-the-art (Mihalcea and Tarau 2004; Wan and Xiao 2008; Liu et al. 2010; Gollapalli and Caragea 2014). These techniques construct a word graph from each target document, in which nodes correspond to words and edges correspond to word association patterns. Nodes are then ranked using graph centrality measures such as PageRank or HITS and the top ranked phrases are returned as keyphrases.

Since the introduction of these techniques, many graph-based extensions have been proposed, which are aimed at modeling various types of information. For example, Wan and Xiao (2008) proposed a model that incorporates a local neighborhood of the target document. Liu et al. (2010) assumed a mixture of topics over documents and used topic models to decompose the topics in order to select keyphrases from all major topics. We posit that other information exists that can be leveraged to improve keyphrase extraction.

Intuitively, keyphrases occur on positions very close to the beginning of a document and they occur frequently. Consider this paper as an example. One representative phrase is "keyphrase extraction." Notice that the phrase occurs very early (even in the paper's title) and occurs frequently. Based on these observations, we propose an unsupervised graph-based model, called PositionRank, that incorporates information from all positions of a word's occurrences into a biased PageRank to score keywords that are later used to score keyphrases. We experimentally evaluate PositionRank on two datasets of research papers and show statistically significant improvements over strong baselines.

Proposed Model

PositionRank involves three essential steps, detailed below.

Graph Construction. Let d be a target document for extracting keyphrases. We build an undirected word graph $G = (V, E)$, where each unique word that passes the part-of-speech filters corresponds to a vertex $v_i \in V$. Two vertices v_i and v_j are linked by an edge $(v_i, v_j) \in E$ if the words corresponding to these vertices co-occur within a window of w contiguous tokens in *d*. The weight of an edge (denote w_{ij}) is computed based on the co-occurrence count of the two words within a window of w successive tokens in d.

Position-Biased PageRank. Let G be an undirected graph constructed as above. We score the vertices in G using a position biased-PageRank algorithm. That is, the score s for vertex v_i is obtained by recursively computing the equation:

$$s(v_i) = \alpha \cdot p(v_i) + (1 - \alpha) \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} s(v_j),$$

where α is a dumping factor, usually set to 0.15, and $p(v_i)$ is a weight assigned to vertex v_i. The term $\alpha \cdot p(v_i)$ is added to ensure that the PageRank algorithm does not get stuck into cycles and can jump to another vertex in the graph with probability $p(v_i)$. In unbiased PageRank, each vertex has equal probability, whereas in biased PageRank, some vertices have higher probability than others (Haveliwala 2003).

The idea of PositionRank is to assign larger weights (or probabilities) to those words that occur very early in a doc-

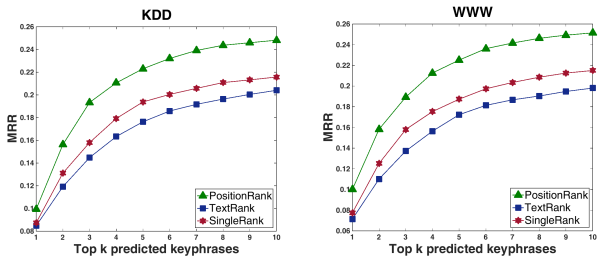

Figure 1: PositionRank vs. unbiased PageRank algorithms.

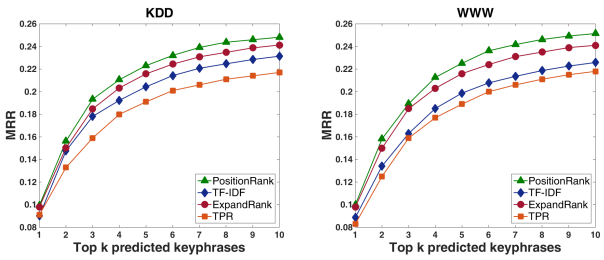

Figure 2: PositionRank vs. strong baselines.

ument. Specifically, we propose to assign a higher probability to a word found on the 2^{nd} position as compared with a word found on the 50^{th} position in the same document. The weight of each candidate word is equal to its inverse position in the document. If the same word appears multiple times in the target document, then we sum all its position weights. For example, a word v_i occurring in the following positions: 2^{nd}, 5^{th} and 10^{th}, has a weight $p(v_i) = \frac{1}{2} + \frac{1}{5} + \frac{1}{10} = \frac{4}{5} = 0.8$. The weights of words are normalized before they are used in the position-biased PageRank.

Forming Candidate Phrases. Candidate words that have contiguous positions in a document are concatenated into phrases and are scored by using the sum of scores of individual words that comprise the phrase (Wan and Xiao 2008).

Experiments and Results

Datasets and Evaluation Measures. To evaluate the performance of PositionRank, we carried out experiments on two datasets. Both datasets were made available by Gollapalli and Caragea (2014).[1] These datasets consist of research papers from the ACM Conference on Knowledge Discovery and Data Mining (KDD) and the World Wide Web Conference (WWW). The author-input keyphrases of a paper were used as gold-standard for evaluation.

To evaluate our model, we report the mean reciprocal rank (MRR), which provides the averaged ranking of the first correct prediction over the set of available documents.

Results and Discussion. We first compare our position-biased PageRank model with two unbiased PageRank models that do not make use of the position information, i.e., TextRank and SingleRank. In TextRank, a document is represented as a word graph according to adjacent words, then PageRank is used to measure the word importance in the document. SingleRank extends TextRank by adding weighted edges between words within a window size greater than 2. Figure 1 shows the MRR curves comparing PositionRank with TextRank and SingleRank. As can be seen, PositionRank substantially outperforms both TextRank and SingleRank on both datasets, illustrating that the words' positions can aid the keyphrase extraction task.

Next, we compare PositionRank with three strong baselines: TF-IDF, ExpandRank (Wan and Xiao 2008), and TopicalPageRank (TPR) (Liu et al. 2010). In TF-IDF, we calculate the *tf* of each candidate word in the target document, whereas the *idf* is estimated from both datasets. In ExpandRank, we use textually-similar neighboring documents to enrich the knowledge in the word graph. In TPR, we run multiple PageRanks on the word graph, one biased PageRank for each topic. We trained the topic model, on a subset of about $45,000$ paper abstracts extracted from CiteSeerX.

The comparison of PositionRank with these baselines in terms of MRR is shown in Figure 2. We can see that PositionRank achieves a significant increase in MRR over the baselines, on both datasets. For example, the highest MRR relative improvement for this experiment is as high as 26.4% achieved on the WWW collection. ExpandRank is clearly the best performing baseline in this experiment, while TPR achieves the lowest MRR values, on our datasets.

Conclusion and Future Work

We proposed a new unsupervised graph-based algorithm, called PositionRank, which incorporates both the relative position and the frequency of a word into a biased PageRank. Our experiments on two datasets show that our proposed model achieves better results than strong baselines, with improvements in performance as high as 26.4%. In future, it would be interesting to evaluate PositionRank on other types of documents, e.g., news articles and political documents.

Acknowledgments. This research is supported by the National Science Foundation award #1423337.

References

Caragea, C.; Bulgarov, F.; Godea, A.; and Gollapalli, S. D. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *EMNLP'14*, 1435–1446.

Gollapalli, S. D., and Caragea, C. 2014. Extracting keyphrases from research papers using citation networks. In *AAAI'14*.

Hasan, K. S., and Ng, V. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *ACL'14*, 1262–1273.

Haveliwala, T. H. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE'03* 784–796.

Kim, S. N.; Medelyan, O.; Kan, M.-Y.; and Baldwin, T. 2012. Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation* 47(3):723–742.

Liu, Z.; Huang, W.; Zheng, Y.; and Sun, M. 2010. Automatic keyphrase extraction via topic decomposition. In *EMNLP'10*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *EMNLP'04*, 404–411.

Wan, X., and Xiao, J. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI'08*, 855–860.

[1] http://www.cse.unt.edu/%7eccaragea/keyphrases.html