Salience Rank: Efficient Keyphrase Extraction with Topic Modeling

Nedelina Teneva*
The University of Chicago
nteneva@uchicago.edu

Weiwei Cheng
Amazon
weiweic@amazon.de

Abstract

Topical PageRank (TPR) uses latent topic distribution inferred by Latent Dirichlet Allocation (LDA) to perform ranking of noun phrases extracted from documents. The ranking procedure consists of running PageRank K times, where K is the number of topics used in the LDA model. In this paper, we propose a modification of TPR, called *Salience Rank*. Salience Rank only needs to run PageRank once and extracts comparable or better keyphrases on benchmark datasets. In addition to quality and efficiency benefits, our method has the flexibility to extract keyphrases with varying tradeoffs between topic specificity and corpus specificity.

1 Introduction

Automatic keyphrase extraction consists of finding a set of terms in a document that provides a concise summary of the text content (Hasan and Ng, 2014). In this paper we consider unsupervised keyphrase extraction, where no human labeled corpus of documents is used for training a classifier (Grineva et al., 2009; Pasquier, 2010; Liu et al., 2009b; Zhao et al., 2011; Liu et al., 2009a). This is a scenario often arising in practical applications as human annotation and tagging is both time and resource consuming. Unsupervised keyphrase extraction is typically casted as a ranking problem – first, candidate phrases are extracted from documents, typically noun phrases identified by part-of-speech tagging; then these candidates are ranked. The performance of unsupervised keyphrase extraction algorithms is evaluated by comparing the most highly ranked keyphrases with keyphrases assigned by annotators.

This paper proposes *Salience Rank*, a modification of Topical PageRank algorithm by Liu et al. (2010). Our method is close in spirit to Single Topical PageRank by Sterckx et al. (2015) and includes it as a special case. The advantages of Salience Rank are twofold:

Performance: The algorithm extracts high-quality keyphrases that are comparable to, and sometimes better than, the ones extracted by Topical PageRank. Salience Rank is more efficient than Topical PageRank as it runs PageRank once, rather than multiple times.

Configurability: The algorithm is based on the concept of "word salience" (hence its name), which is described in Section 3 and can be used to balance topic specificity and corpus specificity of the extracted keyphrases. Depending on the use case, the output of the Salience Rank algorithm can be tuned accordingly.

2 Review of Related Models

Below we introduce some notation and discuss approaches that are most related to ours.

Let $W = \{w_1, w_2, \ldots, w_N\}$ be the set of all the words present in a corpus of documents. Let $G = (W, E)$ denote a word graph, whose vertices represent words and an edge $e(w_i, w_j) \in E$ indicates the relatedness between words w_i and w_j in a document (measured, e.g., by co-occurrence or number of co-occurrences between the two words). The outdegree of vertex w_i is given by $Out(w_i) = \sum_{i:w_i \to w_j} e(w_i, w_j)$.

2.1 Topical PageRank

The main idea behind Topical PageRank (TPR) (Liu et al., 2010) is to incorporate topical information by performing Latent Dirichlet Allocation (LDA) (Blei et al., 2003) on a corpus of documents. TPR constructs a word graph $G = (W, E)$ based on the word co-occurrences within documents. It uses LDA to find the latent topics of the

* Work done as an intern at Amazon.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 530–535
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-2084

document, reweighs the word graph according to each latent topic, and runs PageRank (Page et al., 1998) once per topic.

In LDA each word w of a document d is assumed to be generated by first sampling a topic $t \in T$ (where T is a set of K topics) from d's topic distribution θ^d and then sampling a word from the distribution over words ϕ^t of topic t. Both θ^d and ϕ^t are drawn from conjugate Dirichlet priors α and β, respectively. Thus, the probability of word w, given document d and the priors α and β, is

$$p(w \mid d, \alpha, \beta) = \sum_{t \in T} p(w \mid t, \beta)\, p(t \mid d, \alpha)\,. \quad (1)$$

After running LDA, TPR ranks each word $w_i \in W$ of G by

$$R_t(w_i) = \lambda \sum_{j: w_j \to w_i} \frac{e(w_i, w_j)}{Out(w_j)} R_t(w_j) + (1-\lambda) p(t \mid w_i)\,, \quad (2)$$

for $t \in T$, where $p(t \mid w)$ is estimated via LDA.

TPR assigns a topic specific preference value $p(t \mid w)$ to each $w \in W$ as the jump probability at each vertex depending on the underlying topic. Intuitively, $p(t \mid w)$ indicates how much the word w focuses on topic t.[1]

At the next step of TPR, the word scores (2) are accumulated into keyphrase scores. In particular, for each topic t, a candidate keyphrase is ranked by the sum of the word scores

$$R_t(\text{phrase}) = \sum_{w_i \in \text{phrase}} R_t(w_i)\,. \quad (3)$$

By combining the topic specific keyphrase scores $R_t(\text{phrase})$ with the probability $p(t \mid d)$ derived from the LDA we can compute the final keyphrase scores across all K topics:

$$R(\text{phrase}) = \sum_{t \in T} R_t(\text{phrase})\, p(t \mid d)\,. \quad (4)$$

2.2 Single Topical PageRank

Single Topical PageRank (STPR) was recently proposed by Sterckx et al. (2015). It aims to reduce the runtime complexity of TPR and at the same time maintain its predictive performance. Similar to Salience Rank, it runs PageRank once. STPR is based on the idea of "topical word importance" $TWI(w)$, which is defined as the cosine similarity between the vector of word-topic probabilities $[p(w \mid t_1), \ldots, p(w \mid t_K)]$ and the vector of document-topic probabilities $[p(t_1 \mid d), \ldots, p(t_K \mid d)]$, for each word w given the document d. STPR then uses PageRank to rank each word $w_i \in W$ by replacing $p(t \mid w_i)$ in (2) with $\frac{TWI(w_i)}{\sum_{w_k \in W} TWI(w_k)}$.

STPR can be seen as a special case of Salience Rank, where topic specificity of a word is considered when constructing the random walk, but corpus specificity is neglected. In practice, however, balancing these two concepts is important. It may explain why Salience Rank outperforms STPR in our experiments.

3 Salience Rank

In order to achieve performance and configurability, the Salience Rank (SR) algorithm combines the K latent topics estimated by LDA into a word metric, called *word salience*, and uses it as a preference value for each $w_i \in W$. Thus, SR needs to perform only a single run of PageRank on the word graph G in order to obtain a ranking of the words in each document.

3.1 Word Salience

In the following we provide quantitative measures for topic specificity and corpus specificity, and define word salience.

Definition 3.1 *The topic specificity of a word w is*

$$\begin{aligned} TS(w) &= \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)} \\ &= KL\left(p(t \mid w) \,\|\, p(t)\right)\,. \end{aligned} \quad (5)$$

The definition of topic specificity of a word w is equivalent to Chuang et al. (2012)'s proposal of the *distinctiveness* of a word w, which is in turn equivalent to the Kullback-Leibler (KL) divergence from the marginal probability $p(t)$, i.e., the likelihood that any randomly selected word is generated by topic t, to the conditional probability $p(t \mid w)$, i.e., the likelihood that an observed word w is generated by a latent topic t. Intuitively, topic specificity measures how much a word is shared across topics: The less w is shared across topics, the higher its topic specificity $TS(w)$.

As $TS(w)$ is non-negative and unbounded, we can empirically normalize it to $[0, 1]$ by

$$\frac{TS(w) - \min_u TS(u)}{\max_u TS(u) - \min_u TS(u)}$$

[1] Liu et al. (2010) proposed two other quantities to bias the random walk, $p(w \mid t)$ and $p(w \mid t)p(t \mid w)$, and showed that $p(t \mid w)$ achieves the best empirical result. We therefore adopt the use of $p(t \mid w)$ here.

with the minimum and maximum topic specificity values in the corpus. In what follows, we always use normalized topic specificity values, unless explicitly stated otherwise.

We apply a straightforward definition for corpus specificity.

Definition 3.2 *The corpus specificity of a word w is*

$$CS(w) = p(w \mid corpus) . \quad (6)$$

The corpus specificity $CS(w)$ of a word w can be estimated by counting word frequencies in the corpus of interest. Finally, a word's salience is defined as a linear combination of its topic specificity and corpus specificity.

Definition 3.3 *The salience of a word w is*

$$S(w) = (1-\alpha)CS(w) + \alpha TS(w) , \quad (7)$$

where $\alpha \in [0,1]$ is a parameter controlling the tradeoff between the corpus specificity and the topic specificity of w.

On one hand, we aim to extract keyphrases that are relevant to one or more topics while, on the other hand, the extracted keyphrases as a whole should have a good coverage of the topics in the document. Depending on the downstream applications, it is often useful to be able to control the balance between these two competing principles. In other words, sometimes keyphrases with high topic specificity (i.e., phrases that are representative exclusively for certain topics) are more appropriate, while other times keyphrases with high corpus specificity (i.e., phrases that are representative of the corpus as a whole) are more appropriate. Intuitively, it is advantageous for a keyphrase extraction algorithm to have an internal "switch" tuning the extent to which extracted keyphrases are skewed towards particular topics and, conversely, the extent to which keyphrases generalize across different topics.

It needs to be emphasized that the choice of quantitative measures for topic specificity and corpus specificity used above is just one among many possibilities. For example, for topic specificity, one can make use of the topical word importance by Sterckx et al. (2015), or the several other alternatives mentioned in Section 2.1 proposed by Liu et al. (2010). For corpus specificity, alternatives besides vanilla term frequencies, such as augmented frequency (to discount longer documents) and logarithmically scaled frequency, quickly come into mind.

Taking word salience into account, we modify (2) as follow:

$$R(w_i) = \lambda \sum_{j:w_j \to w_i} \frac{e(w_j, w_i)}{Out(w_j)} R(w_j) + (1-\lambda)S(w_i) . \quad (8)$$

The substantial efficiency boost of SR comparing to TPR lies in the fact that in (2) K PageRanks are required to calculate $R_t(w_i), t = 1 \ldots K$ before obtaining $R(w_i)$, while in (8) $R(w_i)$ is obtained with a single PageRank.

3.2 Algorithm Description

First, SR performs LDA to estimate the latent topics $p(t)$ presented in the corpus and the probability $p(t \mid w)$, which are used to calculate the topic specificity and the salience of each word w.

Similarly to TPR, SR is performed on the word co-occurrence graph $G = (W, E)$. We use undirected graphs: When sliding a window of size s through the document, a link between two vertices is added if these two words appear within the window. It was our observation that the edge direction does not affect the keyphrase extraction performance much. The same observation was noted by Mihalcea and Tarau (2004) and Liu et al. (2010).

We then run the updated version of PageRank derived in (8) and compute the scores of the candidate keyphrases similarly to the way TPR does using (4). For a fair comparison, noun phrases with the pattern `(adjective)*(noun)+` are chosen as candidate keyphrases, which represents zero or more adjectives followed by one or more nouns. It is the same pattern suggested by Liu et al. (2010) in the original TPR paper. SR combines the K PageRank runs in TPR into a single one using salience as a preference value in the word graph.

4 Results

Our experiments are conducted on two widely used datasets in the keyphrase extraction literature, *500N-KPCrowd* (Marujo et al., 2013) and *Inspec* (Hulth, 2003). The *500N-KPCrowd* dataset consists of 500 news articles, 50 stories for each of 10 categories, manually annotated with keyphrases by 20 Amazon Mechanical Turk workers. The *Inspec* dataset is a collection of 2000 paper abstracts of Computer Science & Information Technology journal with manually assigned

dataset	algorithm	precision	recall	F measure
500N-KPCrowd	TPR	0.254	0.222	0.229 (±0.010)
	STPR	0.252	0.221	0.228 (±0.011)
	SR	0.253	0.222	0.229 (±0.010)
Inspec	TPR	0.225	0.255	0.227 (±0.007)
	STPR	0.222	0.254	0.224 (±0.007)
	SR	0.265	0.298	0.266 (±0.007)

Table 1: Comparison of the algorithms on *500N-KPCrowd* and *Inspec*. On both datasets, TPR, STPR and SR were run with 50 LDA topics. In all experiments we used a damping factor $\lambda = 0.85$ in PageRank, as in the original PageRank algorithm, and a window size $s = 2$ to construct the word graphs. Changing the window size s from 2 to 20 does not influence the results much, as also observed in Liu et al. (2010). The convergence of PageRank is achieved when the l^2 norm of the vector containing $R(w_i)$ changes smaller than 10^{-6}. The tradeoff parameter α in SR is fixed at 0.4. The 95% confidence interval for the F measure is shown in the last column.

# topics	precision	recall	F measure
5	0.249	0.218	0.225 (±0.011)
50	0.253	0.222	0.229 (±0.010)
250	0.247	0.216	0.223 (±0.011)
500	0.247	0.216	0.223 (±0.011)

Table 2: Effect of the number of LDA topics when the top 50 keyphrases were used for evaluating SR on *500N-KPCrowd*. The 95% confidence interval for the F measure is shown in the last column.

α	precision	recall	F measure
1.0	0.247	0.216	0.223 (±0.011)
0.7	0.248	0.216	0.223 (±0.011)
0.4	0.248	0.217	0.224 (±0.011)
0.1	0.254	0.222	0.229 (±0.010)
0.0	0.248	0.217	0.224 (±0.011)

Table 3: Effect of the α parameter in SR on *500N-KPCrowd*. SR was run with 50 LDA topics and the top 50 keyphrases were used for the evaluation. The 95% confidence interval for the F measure is shown in the last column.

keyphrases by the authors. Following the evaluation process described in Mihalcea and Tarau (2004), we use only the uncontrolled set of annotated keyphrases for our analysis. Since our approach is completely unsupervised, we combine the training, testing, and validation datasets. Top 50 and 10 keyphrases were used for evaluation on *500N-KPCrowd* and *Inspec*, respectively.[2]

We compare the performance of Salience Rank (SR), Topical PageRank (TPR), and Single Topical PageRank (STPR) in terms of precision, recall and F measure on *500N-KPCrowd* and *Inspec*. The results are summarized in Table 1. Details on parametrization are given in the caption. In terms of the F measure, SR achieves the best results on both datasets. It ties TPR and outperforms STPR on *500N-KPCrowd*, and outperforms both TPR and STPR on *Inspec*. The source code is available at `https://github.com/methanet/saliencerank.git`.

We further experiment with varying the number of topics K used for fitting the LDA model in SR. Table 2 shows how the F measures change on *500N-KPCrowd* as the number of topics varies. Overall, the impact of topic size is mild, with $K = 50$ being the optimal value. The impact of K on TPR can be found in Liu et al. (2010). In our approach, the random walk derived in (8) depends on the word salience, which in turn depends on K; In TPR, not only the individual random walk (2) depends on K, but the final aggregation of rankings of keyphrases also depends on K.

We also experiment with varying the tradeoff parameter α of SR. With *500N-KPCrowd*, Table 3 illustrates that different α can have a considerable impact on various performance measures. To complement the quantitative results in Table 3, Table 4 presents a concrete example, showing that varying α can lead to qualitative changes in the top ranked keyphrases. In particular, when $\alpha = 0$ the corpus specificity of the keyphrases SR extracts is high. This is demonstrated by the fact that words such as "theory" and "function" are among

[2]There are two common ways to set the number of output keyphrases: using a fixed value a priori as we do (Turney, 1999) or deciding a value with heuristics at runtime (Mihalcea and Tarau, 2004).

Input: Individual rationality, or doing what is best for oneself, is a standard model used to explain and predict human behavior, and von Neumann-Morgenstern game theory is the classical mathematical formalization of this theory in multiple-agent settings. Individual rationality, however, is an inadequate model for the synthesis of artificial social systems where cooperation is essential, since it does not permit the accommodation of group interests other than as aggregations of individual interests. Satisficing game theory is based upon a well-defined notion of being good enough, and does accommodate group as well as individual interests through the use of conditional preference relationships, whereby a decision maker is able to adjust its preferences as a function of the preferences, and not just the options, of others. This new theory is offered as an alternative paradigm to construct artificial societies that are capable of complex behavior that goes beyond exclusive self interest.	
Unique top keyphrases with $\alpha = 0$: classical mathematical formalization preferences theory options function multiple agent settings	**Unique top keyphrases with $\alpha = 1$:** individual interests group interests artificial social systems individual rationality conditional preference relationships standard model

Table 4: An example of running SR on an *Inspec* abstract with a minimum and maximum value of α. Unique keyphrases among the top 10 are shown.

the top keyphrases SR selects, which are highly common words in scientific papers. On the other hand, when $\alpha = 1$ these keyphrases are not presented among the top. This toy example illustrates the relevance of balancing topic and corpus specificity in practice: When presenting the keyphrases to a layman, high corpus specificity is suitable as it conveys more high-level information; when presenting to an expert in the area, high topic specificity is suitable as it dives deeper into topic specific details.

5 Conclusions & Remarks

In this paper, we propose a new keyphrase extraction method, called Salience Rank. It improves upon the Topical PageRank algorithm by Liu et al. (2010) and the Single Topical PageRank algorithm by Sterckx et al. (2015). The key advantages of this new method are twofold: (i) While maintaining and sometimes improving the quality of extracted keyphrases, it only runs PageRank once instead of K times as in Topical PageRank, therefore leads to lower runtime; (ii) By constructing the underlying word graph with newly proposed word salience, it allows the user to balance topic and corpus specificity of the extracted keyphrases.

These three methods rely only on the input corpus. They can be benefited by external resources like Wikipedia and WordNet, as indicated by, e.g., Medelyan et al. (2009), Grineva et al. (2009), Martinez-Romo et al. (2016).

In the keyphrase extraction literature, LDA is the most commonly used topic modeling method. Other methods, such as probabilistic latent semantic indexing (Hofmann, 1999), nonnegative matrix factorization (Sra and Inderjit, 2006), are viable alternatives. However, it is hard to tell in general if the keyphrase quality improves with these alternatives. We suspect that strongly depends on the domain of the dataset and a choice may be made depending on other practical considerations.

We have fixed the tradeoff parameter α throughout the experiments for a straightforward comparison to other methods. In practice, one should search the optimal value of α for the task at hand. An open question is how to theoretically quantify the relationship between α and various performance measures, such as the F measure.

Acknowledgments

We would like to thank Matthias Seeger, Cedric Archambeau, Jan Gasthaus, Alex Klementiev, Ralf Herbrich, and the anonymous ACL reviewers for their valuable inputs.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3(1):993–1022.

Jason Chuang, Christopher D Manning, and Jeffrey Heer. 2012. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. pages 74–77.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th International Conference on World Wide Web*. WWW, pages 661–670.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. ACL, pages 1262–1273.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR, pages 50–57.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. EMNLP, pages 216–223.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009a. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. NAACL, pages 620–628.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. EMNLP, pages 366–376.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language*. EMNLP, pages 257–266.

Juan Martinez-Romo, Lourdes Araujo, and Andres Duque Fernandez. 2016. Semgraph: Extracting keyphrases following a novel semantic graph-based approach. *Journal of the Association for Information Science and Technology* 67(1):71–82.

Luís Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P Neto. 2013. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. *arXiv preprint arXiv:1306.4886* .

Olena Medelyan, Eibe Frank, and Ian Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. EMNLP, pages 1318–1327.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. EMNLP, pages 404–411.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University.

Claude Pasquier. 2010. Single document keyphrase extraction using sentence clustering and latent Dirichlet allocation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. pages 154–157.

Suvrit Sra and Dhillon Inderjit. 2006. Generalized nonnegative matrix approximations with Bregman divergences. In *Advances in Neural Information Processing Systems 18*. NIPS, pages 283–290.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web*. WWW, pages 121–122.

Peter Turney. 1999. Learning to extract keyphrases from text. Technical report, National Research Council Canada, Institute for Information Technology.

Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. ACL, pages 379–388.