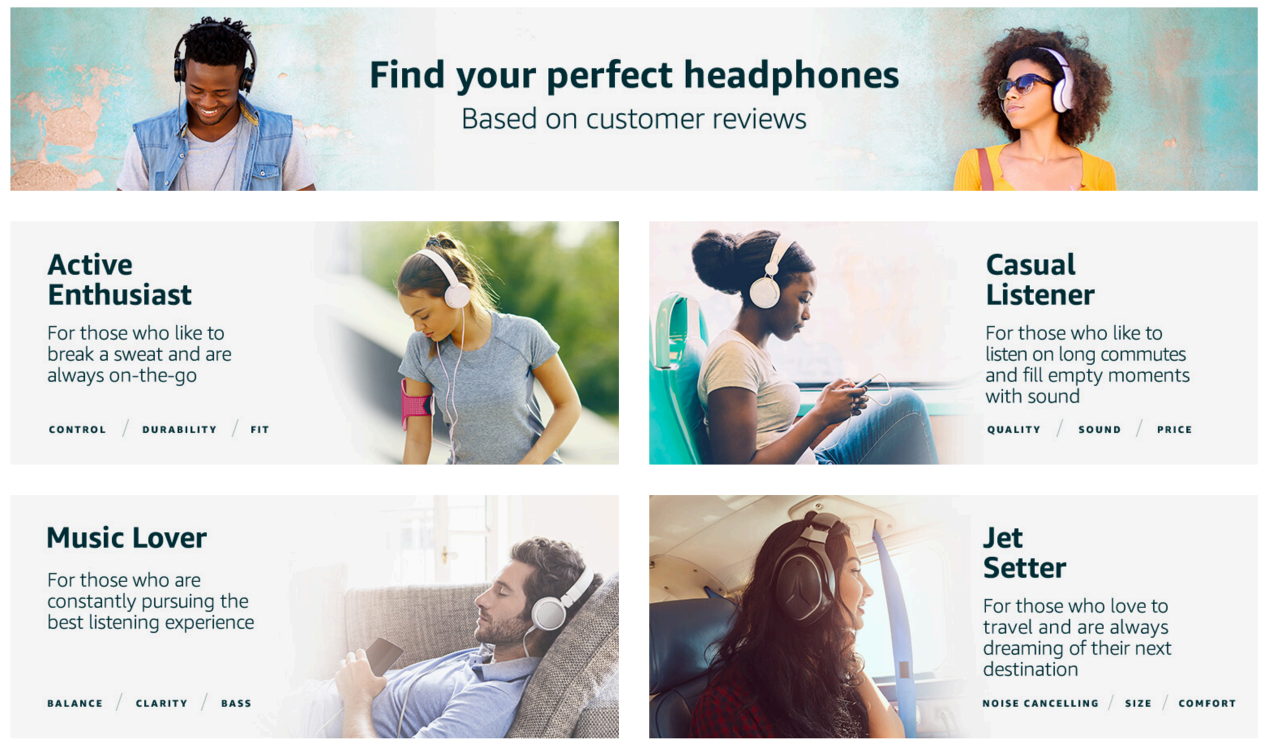
Salience Rank

Efficient Keyphrase Extraction with Topic Modeling

Nedelina Teneva
The University of Chicago

Weiwei Cheng
amazon

(1) Introduction



Keyphrase extraction aims to find a collection of phrases in a document that provides a concise summary of the text content.

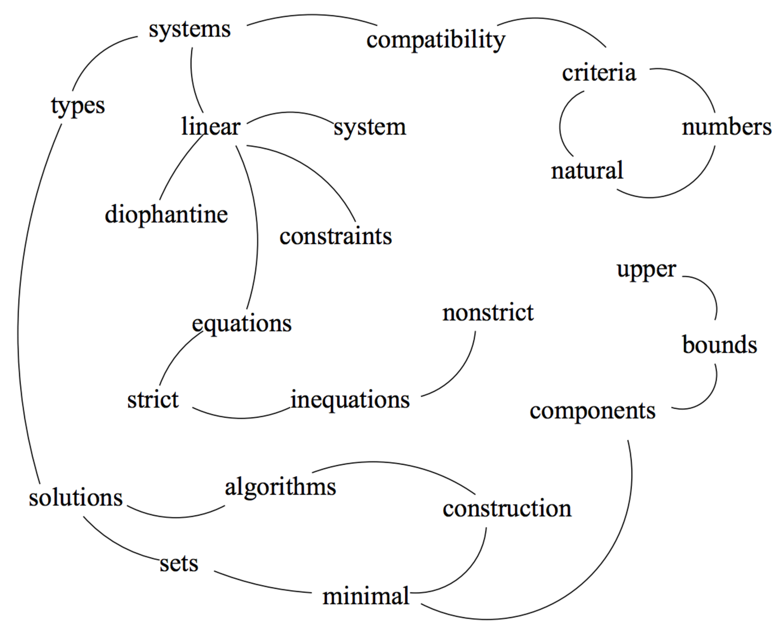
- **Inputs**: a text document
- **Outputs**: a set/ranking of phrases
- **Evaluation** is done by comparing to human annotated keyphrases via measures such as *precision*, *recall*, *F score*, etc.

(2) Overview

An automatic keyphrase extraction system typically operates in 2 steps:

1. Extract a list of phrases as **candidate phrases** with some heuristics.
   - Noun phrases with `(adjective)*(noun)+`
   - Phrases that don't contain predefined stopwords
   - etc.
2. Select keyphrases from these candidates with **supervised** or **unsupervised** approaches.
   - Supervised: binary classification (Frank et al. 1999), pairwise ranking (Jiang et al. 2009)
   - Unsupervised: graph-based ranking (Mihalcea & Tarau, 2004), topic-based clustering (Grineva et al., 2009), language modeling (Tomokiyo & Hurst, 2003)

(3) Topical PageRank (Liu et al., 2010)



Intuition: A candidate keyphrase is important if it is related to other candidates, which in turn also have high importance.

Overall Procedure:

1. Build a word graph from input document.
2. Perform random walk to obtain word scores.
3. Select keyphrases with word scores.

Concrete Procedure:

- Given a word graph $G = (W, E)$, where an edge $e(w_i, w_j)$ indicates relatedness between w_i and w_j, the score of each word w_i under topic $t \in T$ is determined by

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

  where $Out(w_i) = \sum_{i:\, w_i \to w_j} e(w_i, w_j)$ is the outdegree of vertex w_i, and $p(t \mid w_i)$ is a topic specific jump probability of w_i, derived from LDA.
- Then for topic t, we obtain keyphrase scores $R_t(\text{phrase}) = \sum_{w_i \in \text{phrase}} R_t(w_i)$.
- The final keyphrase scores are given by $R(\text{phrase}) = \sum_{t \in \mathrm{T}} R_t(\text{phrase})\, p(t \mid \mathrm{d})$.

(4) Salience Rank

Performance: While still exploiting the structure information derived by LDA, we run PageRank once instead of K times and achieve similar keyphrase quality.

Configurability: Users can balance *topic specificity* and *corpus specificity* of the extracted keyphrases and can tune the results according to particular use cases.

- On one hand, we aim to extract keyphrases that are relevant to specific topics;
- On the other hand, the extracted keyphrases as a whole should have a good coverage of the major topics in the document.
- It is often useful to control the balance between these two competing principles.

Definitions:

- The *topic specificity* of a word w: $TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$
- The *corpus specificity* of a word w: $CS(w) = p(w \mid \text{corpus})$
- The *salience* of a word w: $S(w) = (1-\alpha)\, CS(w) + \alpha\, TS(w)$, where α is the tradeoff parameter balancing corpus and topic specificity of w.

Our random walk:

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

Comparing to TPR, PageRank needs to be run **only once**.

(5) Experiments

dataset	algorithm	precision	recall	F score
500news	TPR	0.254	0.222	0.229 (±0.010)
	SR	0.253	0.222	0.229 (±0.010)
Inspec	TPR	0.225	0.255	0.227 (±0.007)
	SR	0.265	0.298	0.266 (±0.007)

- In terms of **performance**, while computationally more efficient, Salience Rank obtains comparable or better keyphrases on benchmark data.

α	precision	recall	F score
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)

500news

Unique top keyphrases when $\alpha = 0$	Unique top keyphrases when $\alpha = 1$
classical mathematical formalization	individual interests
preferences	group interests
theory	artificial social systems
options	individual rationality
function	conditional preference relationships
multiple agent settings	Neumann-Morgenstern theory

on one Inspec abstract

- In terms of **configurability**: (1) Balancing TS and CS considerably impacts results; (2) Qualitatively, high CS tends to be good for a layman and high TS good for an expert.

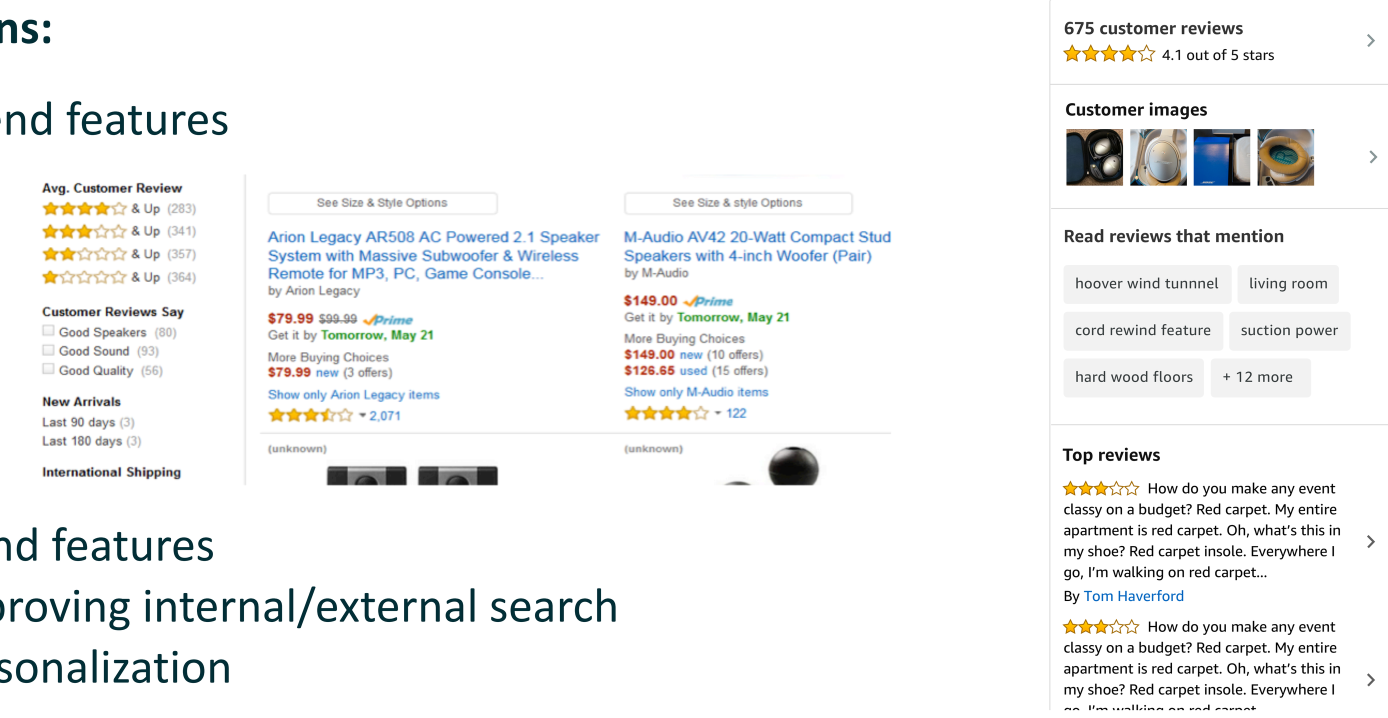
(6) Conclusions

We proposed an unsupervised keyphrase extraction algorithm, **Salience Rank**, that improves the state-of-the-art.

- **Performance**: While still exploiting the structure information derived by LDA, we run PageRank only once and obtain similar or better keyphrases.
- **Configurability**: Users can balance topic specificity and corpus specificity of the extracted keyphrases and can tune the results according to use cases.

Applications:

- Frontend features
- Backend features
  - Improving internal/external search
  - Personalization
  - etc.



weiweic@amazon.com